INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Human Motion Tracking with a Kinematic Parameterization of Extremal Contours

David Knossow, Rémi Ronfard and Radu Horaud

N° 6007

Octobre 2006

Thème COG

ISSN 0249-6399 ISRN INRIA/RR--6007--FR+ENG

Human Motion Tracking with a Kinematic Parameterization of Extremal Contours

David Knossow, Rémi Ronfard and Radu Horaud

Thème COG — Systèmes cognitifs

Projet MOVI

Rapport de recherche n° 6007 — Octobre 2006 — 35 pages

Abstract: This report addresses the problem of human motion tracking from image sequences. The human body is described by an articulated mechanical chain and human body-parts are described by volumetric primitives with curved surfaces. An extremal contour appears in an image whenever a curved surface turns smoothly away from the viewer. We describe a method that relies on a kinematic parameterization of such extremal contours. The apparent motion of these contours in the image plane is a function of both the rigid motion of the surface and the relative position and orientation of the viewer with respect to the curved surface. First, we describe a parameterization of an extremal-contour point, and its associated image velocity, as a function of the motion parameters of the kinematic chain associated with the human body. Second, we introduce the zero-reference kinematic model and we show how it may be used for human-motion modelling. Third, we show how the chamfer-distance may be used to measure the discrepancy between predicted extremal contours and observed image contours; Moreover we show how the chamfer distance can be used as a differentiable multi-valued function and how the tracker based on this distance can be cast in an optimization framework. Fourth, we describe a practical human-body tracker that may use an arbitrary number of cameras. One great methodological and practical advantage of our method is that it relies neither on model-to-image, nor on image-to-image point matches. In practice we model people with 5 kinematic chains, 19 volumetric primitives, and 54 degrees of freedom; We observe silhouettes in images gathered with several synchronized and calibrated cameras. The tracker has been successfully applied to several complex motions gathered at 30 frames/second.

Key-words: articulated motion representation, human-body tracking, zero-reference kinematics, developable surfaces, extremal contours, chamfer distance, chamfer matching.

This work was partially supported by the RIAM project SEMOCAP.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier (France)
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Suivi du mouvement humain à l'aide de la paramétrisation cinématique des contours extrémaux

Résumé : Dans ce rapport de recherche, nous nous intéressons au problème du suivi du mouvement humain à l'aide de plusieurs caméras. Le modèle que nous utilisons pour effectuer le suivi est décrit par une chaîne articulée ainsi que par des primitives solides. Les surfaces de ces primitives se projettent dans les images sous la forme de deux types de contours : les contours de discontinuité, qui apparaissent à l'intersection entre deux surfaces, et les contours extrémaux qui apparaissent lorsque la surface est tangente au rayon de vue. Dans ce rapport, nous décrivons une méthode de capture du mouvement basé sur l'observation des contours extrémaux. Pour cela, nous introduisons une paramétrisation explicite des contours extrémaux, applicable aux surfaces développables. Cette paramétrisation nous permet de relier le mouvement apparent des contours aux mouvements articulaires et aux positions et orientations des caméras.

Dans un premier temps, nous explicitons la paramétrisation des contours extrémaux et donnons leur vitesse apparente dans les images en fonction des paramètres de la chaîne cinématique auquel la primitive est attachée. Dans un deuxième temps, nous introduisons la modélisation en référence zéro de la chaîne cinématique. Nous montrons comment intégrer cette modélisation dans celle de la chaîne cinématique humaine. Dans un troisième temps, nous montrons comment utiliser la distance de chanfrein pour déterminer l'erreur entre les contours extraits de l'image et les contours extrémaux du modèle. De plus, nous montrons que cette distance peut être différentiable et donc que nous pouvons aborder le problème du suivi de mouvement comme un simple problème de minimisation non-linéaire.

Un avantage majeur tant du point de vue méthodologique que du point de vue pratique est que l'algorithme n'utilise ni de correspondances modèle à image, ni de correspondances image à image. En pratique, nous modélisons le corps humain avec 5 chaînes cinématiques, 19 cônes tronqués à base elliptique et 54 degrés de liberté. Nous observons les silhouettes et le contours dans les images acquises avec des caméras synchronisées et calibrées. L'algorithme de suivi a pu être testé avec succès sur des mouvements complexes.

Mots-clés : Mouvement articulé, suivi du mouvement humain, modélisation en référence zéro, surfaces développables, contours extrémaux, distance de chanfrein.

1 Introduction and background

In this report we address the problem of tracking complex articulated motions from images. The problem of articulated motion (such as human-body motion) representation and tracking from 2-D and 3-D visual data has been thoroughly addressed in the recent past. The problem is difficult because it needs to solve an inverse kinematic problem, namely the problem of finding the parameters characterizing the control space (the space spanned by the articulated parameters) from a set of measurements performed in the observation space. In general this problem cannot be solved explicitly because the dimensionality of the observation space is much smaller than the dimensionality of the control space. More formally, the problem can be stated as the following minimization problem:

$$\min_{\mathbf{\Phi}} \; E(\mathcal{Y}, \mathcal{X}(\mathbf{\Phi})) \tag{1}$$

where $\mathcal{Y}$ denotes a set of observations, $\mathcal{X}$ denotes a set of predictions using the direct kinematic model, and $\mathbf{\Phi}$ is the vector of motion parameters to be estimated.

In this report we will embed the human-motion tracking into the minimization problem defined by eq. (1). We will emphasize a human-body model composed of articulated mechanical chains and of rigid body parts. Each such part is defined as a developable surface. An intrinsic property of such a surface is that it projects onto an image as a pair of straight extremal contours (an extremal contour appears in an image whenever a curved surface turns smoothly away from the viewer). We develop a direct kinematic representation of extremal contours based on the differential properties of developable surfaces and on the zero-reference kinematic representation of articulated chains with rotational joints. This kinematic description encapsulates the constrained articulated motions as well as a free rigid motion and allows us to predict both the position and the velocity of extremal contours.

Therefore, human motion tracking may be formulated as the problem of minimizing eq. (1) using the chamfer distance between predicted extremal contour points, $\mathcal{X}(\mathbf{\Phi})$ and contour points detected in images, $\mathcal{Y}$. We show how the chamfer distance can be used as a differentiable multi-valued function and how the tracker based on this distance can be cast into a non-linear optimization framework. Even if, in theory, one camera may be sufficient for recovering the motion parameters, we show that a multiple-camera setup brings in the necessary robustness for implementing the tracker.

There is, of course, a substantial literature on articulated motion tracking driven by image features and using either a single video or mutliple video sequences.

Monocular approaches generally require a probabilistic framework such as in [7, 27, 26, 1] to cite just a few. The probabilistic formulation has the attraction that both prior knowledge and uncertainty in the data are handled in a systematic way. The first difficulty with these methods is that the image data must be mapped onto a vector space such that statistical methods can be easily applied, which is not a straightforward task in itself. The second difficulty is to establish a relationship (between articulated-motion space and this vector space) that must be learnt prior to tracking. This is not an obvious task because it is virtually impossible to scan in advance the space of all the complex motions of an articulated object with many degrees of freedom. Other methods attempted

to recover articulated motion from image cues such as optical flow through sophisticated non-linear minimization methods [4, 24, 25].

A second class of approaches relies on multiple-video sequences gathered with finely synchronized cameras. These approaches generally use either image contours (extracted with edge detection) or silhouettes (extracted with background subtraction) [12, 16, 6]. These methods use generic models, such as superquadrics, quadrics or simple cylinders, to represent body parts. Nevertheless, projecting these models onto images and comparing them with contours and/or silhouettes is not an obvious task. Previous methods have not made explicit the analytic representation allowing the mapping of articulated objects (and their surfaces) onto silhouettes. Formally, a silhouette is the *occluding contour* [2] that separates an object from the background. Occluding contours are built up of *discontinuity* and *extremal* contours. The former correspond to sharp edges or to surface discontinuities. The latter occur where a curved surface turns smoothly away from the viewer. Notice that the bottom-up segmentation of a silhouette into discontinuity and extremal contours is not an easy task [14].

In the case of sharp edges there are well documented methods allowing for an explicit (analytic) representation of the mapping between the object's constrained (articulated) motion parameters and the observed image contours both under ortography [4] and under perspective projection [9, 10, 19]. In the presence of smooth surfaces, an extremal contour is the projection of a *contour generator* – a virtual contour that lies onto the surface where the lines of sight are tangent to the surface. Therefore, the apparent image motion of an extremal contour is a function of both the motion of the object itself and the motion of the contour generator, the latter being a function of the relative position of the object's surface with respect to the viewer. It turns out that the link between the differential properties of surfaces and the rigid motion of these surfaces has barely been addressed.

In this report we take an approach to human-body tracking based on fitting articulated shapes to image contours. See Figure 1 for an example on how the method operates in practice. We use *elliptical cones* to model body parts. These shapes belong to a more general class of developable surfaces that have interesting differential properties that have not been fully exploited in the past. Elliptical cones in particular and developable surfaces in general project onto images as a set of straight lines. By deliberately considering only these contours we simplify both the tasks of interpreting the image contours and of comparing them to the predicted object contours. Moreover, the body parts are joined together to form an articulated structure composed of five *open kinematic chains*. Therefore, each body-part motion is composed of two motions: a motion constrained by a number of rotational joints (the motion of its associated kinematic chain) and a free motion, i.e., the motion of the root body-part with respect to a world reference frame. We derive an analytic expression for the motion of a predicted extremal-contour point as a function of both the body-part motion as well as the motion of its contour generator lying onto the curved surface of that body part.

Therefore, the problem of articulated motion tracking may be formulated as the problem of minimizing a metric between image contours (gathered simultaneously with several cameras) and extremal contours (predicted from the model). There are several ways of defining a distance between two contours, including the sum of squares of the point-to-point distances, the Hausdorff distance, the chamfer distance, and so forth. We decided to capitalize onto the chamfer distance, and unlike

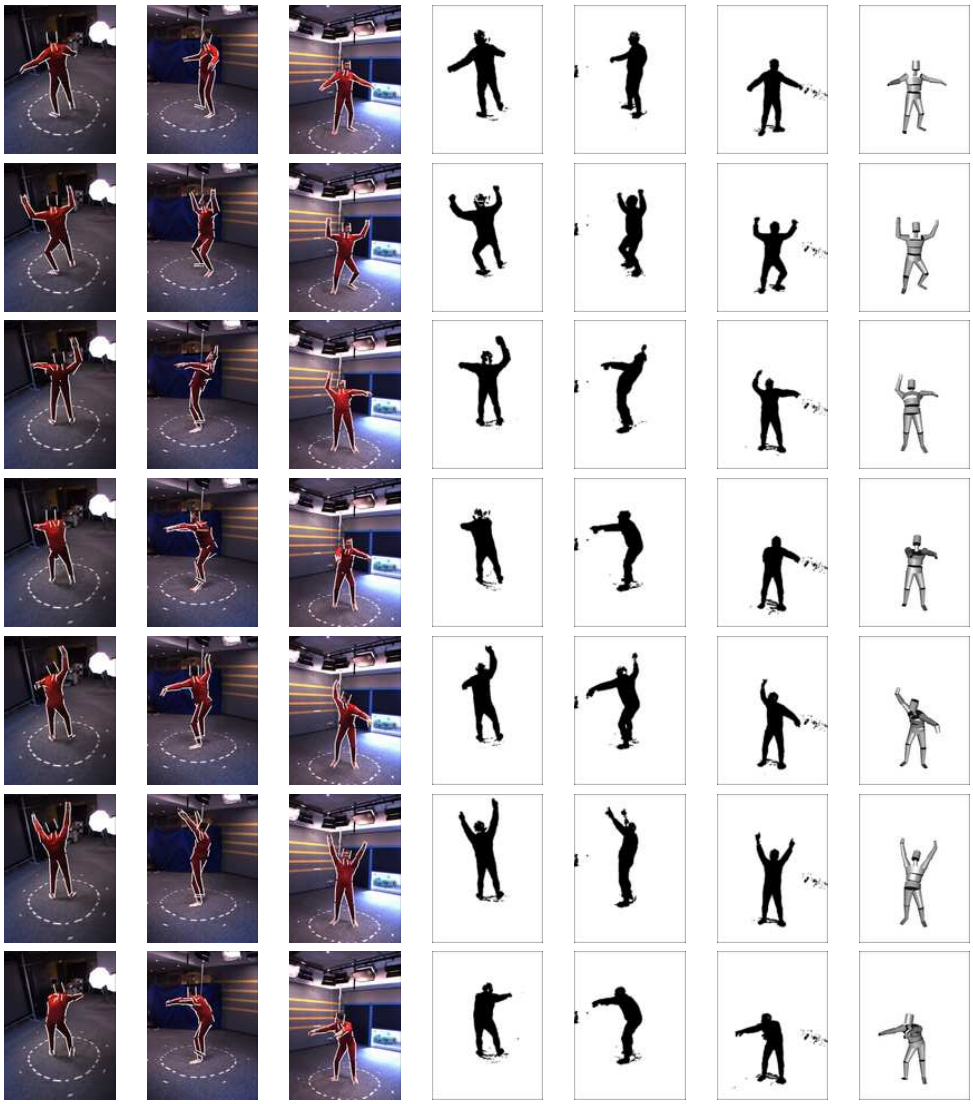


Figure 1: An example of human-motion multiple-camera tracking based on extremal contours. The tracker uses six images sequences out of which only three are shown on this figure. The extremal contours fitted to the image data are shown superimposed onto the raw images. The tracker uses the edges associated with the silhouettes to fit the parameterized extremal contours to the data. The recovered pose of the human-body model is shown from the viewpoint of the third camera. There is a total of 250 images in these sequences. Notice that this apparent simple gesture (raising the arms and then leaning forward) involves almost all the degrees of freedom of the model as well as a motion of the root body-part.

previous approaches, we developed an analytic expression allowing us to compare the real-valued contour points predicted from the model with the chamfer-distance image computed from boolean-valued image contours. This image-to-model metric thus defined does not require point-to-point matches, its computation is very efficient, and it can be analytically differentiated. We analyse in detail the numerical conditionning of the tracker, which amounts to the rank analysis of the Jacobian associated with the direct kinematic model. Although, in principle, one camera may be sufficient for gathering enough data, we claim that a multiple-camera setup provides the redundancy that is absolutely necessary for robust tracking.

Paper organization. The remainder of this report is organized as follows. In section 2 we consider the case of developable surfaces and we show that their contour generators are rulings of the surface. We derive a closed-form solution for the velocity of the contour generators (and of the corresponding extremal contours) as a function of the kinematic screw associated with the motion of the surface. In section 3 we develop an explicit solution for the human-body kinematics using the zero-reference kinematic model and in section 4 we derive the Jacobian that maps joint and free-motion velocities onto the 2-D velocity of an extremal-contour point. Section 5 describes in detail how to fit predicted extremal contours to detected image contours and how to carry out the minimization process using the chamfer distance. Section 6 describes experiments performed with simulated and real data. Finally, section 7 draws some conclusions and give directions for future work.

2 The kinematics of extremal contours

2.1 Definitions and notations

As already explained, we use shapes with smooth surfaces in order to represent rigid body parts. Each such body-part is linked to a *root* body-part trhough a kinematic chain of body parts. Each joint in the kinematic chain – the link between two adjacent body parts – is modelled by a rotational joint. Each such joint may have one, two, or three degrees of freedom. Moreover, the root body-part is allowed to freely move in the 3-D space with six degrees of freedom (three rotations and three translations).

Therefore, the motion of any part of the kinematic chain is obtained by a combination of a *constrained motion* and of a *free motion*. We denote by $\mathbf{\Phi} = (\phi_1, \ldots, \phi_n)$ all these motion parameters. The first q parameters correspond to the motion of the root with $q \leq 6$ and the remaining p parameters correspond to the joint angles: $n = q + p$. The kinematic parameterization will be made explicit in the next section. In this section we will describe the motion of a body-part by a 3×3 rotation matrix $\mathbf{R}$ and by a 3-D translation vector $\boldsymbol{t}$. Both these rotation and translation are in turn parameterized by $\mathbf{\Phi}$, i.e., we will have $\mathbf{R}(\mathbf{\Phi})$ and $\boldsymbol{t}(\mathbf{\Phi})$.

It will also be convenient to consider the pose and the motion of a body part as a rigid object in its own right. A minimal representation of a 3-D rotation uses 3 parameters: Euler angles, unit

vector and angle, etc. Therefore, the *pose of a rigid object* is described by six parameters – three for the rotation and three for the translation – and let $\boldsymbol{r}$ be the pose vector associated with $(\mathbf{R}, \boldsymbol{t})$. If a body-part is treated as a free-moving rigid body, then the 6 components of $\boldsymbol{r}$ are the free parameters. If a body-part is treated as a component of a kinematic chain, $\boldsymbol{r}$ is parameterized by $\boldsymbol{\Phi}$, i.e., $\boldsymbol{r}(\boldsymbol{\Phi})$. Finally we denote by $\dot{\boldsymbol{x}}$ the time derivative of $\boldsymbol{x}$.

We consider now the smooth surface of a body-part. This surface projects onto the image as an extremal contour. The apparent image motion of such an extremal contour depends on the motion of the body-part and on the local shape of the surface. Indeed, let's consider the *contour generator* that lies onto the smooth surface – the locus of points where the surface is tangent to the lines of sight originating from the camera's center of projection. When the surface moves, the contour generator moves as well and it's motion is constrained both by the rigid motion of the surface and by the relative position of the surface with respect to the camera. Therefore, the contour generator has two motion components and we must explicitly estimate these two components.

First, we will determine the constraints that formally define the contour generator. The extremal contour is simply determined by projecting the contour generator onto the image plane. Second, we will derive a closed-form solution for the *extremal-contour Jacobian*, i.e., the Jacobian matrix that maps 3-D joint velocities onto 2-D contour-point velocities.

2.2 The contour-generator constraint and extremal contours

Let X be a 3-D point that lies onto the smooth surface of a body part, and let $\boldsymbol{X} = (X_1, X_2, X_3)$ be the coordinates of this point in the body-part frame, Figure 2. Without loss of generality, the camera frame will be chosen to be identical to the world frame. Hence, the world coordinates of X are:

$$\boldsymbol{X}^w = \mathbf{R}(\boldsymbol{\Phi})\boldsymbol{X} + \boldsymbol{t}(\boldsymbol{\Phi}) \tag{2}$$

The contour generator is the locus of points lying onto the surface where the lines of sight (originating at the optical center of the camera and passing through image points) are tangent to that surface. This tangency is expressed by the dot-product between the surface normal and the line of sight at the point. Nevertheless both the line of sight and the surface normal must be expressed in a common reference frame, and without loss of generality we express these entities in the world (or camera) reference frame. To conclude, the visibility equation that defines the contour generator writes:

$$(\mathbf{R}\boldsymbol{n})^\top (\mathbf{R}\boldsymbol{X} + \boldsymbol{t} - \boldsymbol{C}) = 0 \tag{3}$$

where the surface normal $\boldsymbol{n}$ is defined by the following cross-product:

$$\boldsymbol{n} = \frac{\partial \boldsymbol{X}}{\partial z} \times \frac{\partial \boldsymbol{X}}{\partial \theta} = \boldsymbol{X}_z \times \boldsymbol{X}_\theta \tag{4}$$

Here the couple (z, θ) is a parameterization of the body-part's surface and $\boldsymbol{C}$ is the camera's optical center. The equation above becomes:

$$\boldsymbol{X}^\top \boldsymbol{n} + (\boldsymbol{t} - \boldsymbol{C})^\top \mathbf{R} \boldsymbol{n} = 0 \tag{5}$$

or:

$$(\boldsymbol{X} + \boldsymbol{m})^\top \boldsymbol{n} = 0 \tag{6}$$

with $\boldsymbol{m} = \mathbf{R}^\top (\boldsymbol{t} - \boldsymbol{C})$. Eq. (6) is the contour-generator constraint and it yields at each time instant.

It will be quite difficult to solve this equation in the general case and obtain an explicit description of the contour generator $\boldsymbol{X}(z, \theta)$ for any class of surfaces. Once the contour generator is determined, the 2-D extremal contour (the projection of the contour generator) can be found in the camera frame using perspective projection:

$$\begin{pmatrix} s\boldsymbol{x} \\ s \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{X}^w \\ 1 \end{pmatrix} \tag{7}$$

2.3 The contour generator of a developable surface

As we just outlined, it would be difficult to treat the general case of curved surfaces. An interesting case is the class of developable surfaces which are a special case of ruled surfaces [8]. We prove the following result:

Proposition 1 Under perspective projection, the contour generators of a developable surface are rulings of the surface, i.e., they are line segments.

This also means that the extremal contours of a developable surface are straight lines. In practice we need to consider surfaces that are well suited to model body parts. We will use elliptical cones but the result of this section allows one to use any kind of developable surfaces.

Consider a differentiable one-parameter family of straight lines $(\boldsymbol{\alpha}(\theta), \boldsymbol{\beta}(\theta))$ where to each θ are assigned a 3-D point $\boldsymbol{\alpha}(\theta)$ and a 3-D vector $\boldsymbol{\beta}(\theta)$, so that both $\boldsymbol{\alpha}(\theta)$ and $\boldsymbol{\beta}(\theta)$ depend differentiably on θ. The parametrized surface:

$$\boldsymbol{X}(\theta, z) = \boldsymbol{\alpha}(\theta) + z\, \boldsymbol{\beta}(\theta) \tag{8}$$

is called a ruled surface. Let us compute the Gaussian curvature, K of a ruled surface. A developable surface is a ruled surface such that $K = 0$ at each surface point. The Gaussian curvature of a parameterized surface $\boldsymbol{X}(\theta, z)$ can be computed from the coefficients of its first and second fundamental forms:

$$I(d\theta, dz) = d\boldsymbol{X} \cdot d\boldsymbol{X} = g_{\theta\theta}(d\theta)^2 + g_{z\theta} d\theta dz + g_{zz}(dz)^2 \tag{9}$$

$$II(d\theta, dz) = d^2\boldsymbol{X} \cdot \boldsymbol{n} = L_{\theta\theta} d\theta^2 + L_{z\theta} d\theta dz + L_{zz} dz^2 \tag{10}$$

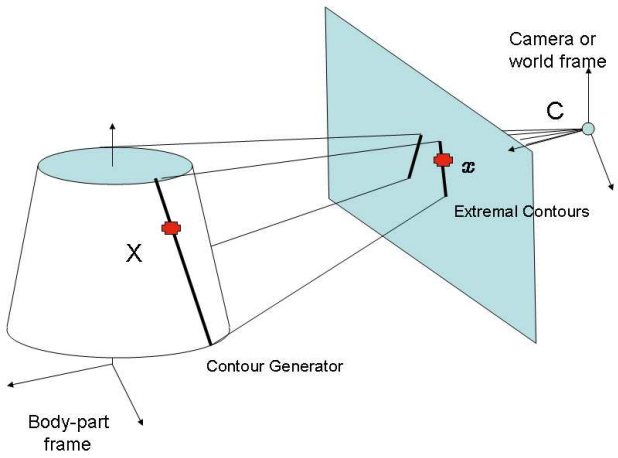


Figure 2: A truncated elliptical cone is an example of a developable surface used to model a body part. Such a surface projects onto an image as a pair of *extremal contours*. The 2-D motion of these extremal contours is a function of both the motion of the body-part itself as well as the sliding of the *contour generator* along the smooth surface of the part.

The Gaussian curvature is given by the formula:

$$K = \frac{L_{\theta\theta}L_{zz} - L_{z\theta}^2}{g_{\theta\theta}g_{zz} - g_{z\theta}^2} \tag{11}$$

The first and second derivatives of a ruled surface are given by:

$$\begin{aligned}
\boldsymbol{X}_\theta &= \boldsymbol{\alpha}'(\theta) + z\,\boldsymbol{\beta}'(\theta) \\
\boldsymbol{X}_z &= \boldsymbol{\beta}(\theta) \\
\boldsymbol{X}_{\theta\theta} &= \boldsymbol{\alpha}''(\theta) + z\,\boldsymbol{\beta}''(\theta) \\
\boldsymbol{X}_{z\theta} &= \boldsymbol{\beta}'(\theta) \\
\boldsymbol{X}_{zz} &= 0
\end{aligned}$$

Notice that the normal to a ruled surface is given by:

$$\boldsymbol{n} = \boldsymbol{X}_\theta \times \boldsymbol{X}_z = (\boldsymbol{\alpha}' + z\,\boldsymbol{\beta}') \times \boldsymbol{\beta} \tag{12}$$

Therefore, the coefficients of the second fundamental form are:

$$\begin{aligned}
L_{\theta\theta} &= (\boldsymbol{\alpha}''(\theta) + z\,\boldsymbol{\beta}''(\theta))^\top (\boldsymbol{\alpha}'(\theta) + z\,\boldsymbol{\beta}'(\theta)) \times \boldsymbol{\beta} \\
L_{z\theta} &= \det([\ \boldsymbol{\alpha}' \quad \boldsymbol{\beta} \quad \boldsymbol{\beta}'\ ]) \\
L_{zz} &= 0
\end{aligned}$$

Hence, the condition that the surface is developable (the Gaussian curvature K must be null everywhere onto the surface) is reduced to the constraint $L_{z\theta} = 0$ (one may easily check that the denominator of eq. (11) is never equal to zero):

$$\det([\ \boldsymbol{\alpha}' \quad \boldsymbol{\beta} \quad \boldsymbol{\beta}'\ ]) = 0$$

This constraint yields that $\boldsymbol{\alpha}'$ may be written as a linear combination of the two other vectors, i.e. $\boldsymbol{\alpha}' = a\boldsymbol{\beta} + b\boldsymbol{\beta}'$. From eq. (12) we obtain a new expression for the normal to the surface:

$$\boldsymbol{n} = (1 + bz)\,\boldsymbol{\beta}' \times \boldsymbol{\beta} \tag{13}$$

Notice that the direction of the normal is given by the cross-product of $\boldsymbol{\beta}'$ and $\boldsymbol{\beta}$ and it depends only on the parameter θ. Using this parameterization of the normal, we can rewrite the contour generator constraint, eq. (6) for developable surfaces as follows:

$$(\boldsymbol{\alpha}(\theta) + \boldsymbol{m})^\top (\boldsymbol{\beta}'(\theta) \times \boldsymbol{\beta}(\theta)) = 0 \tag{14}$$

One should notice that this contour-generator constraint involves only the surface parameter θ and not the z parameter. Therefore, any solution of eq. (14), say $\hat{\theta}$, will yield the entire ruling line $\boldsymbol{X}(\hat{\theta}, z)$. This proves that under perspective projection, the contour generators of a developable surface are rulings of the surface, i.e. line segments. As a result, the kinematics of the contour generators are fully determined by the evolution of the solutions $\hat{\theta}(t)$ of the contour generator equation over time.

2.4 Truncated elliptical cones

In practice will model body parts with truncated elliptical cones. Such a shape is bounded by two planar faces which produce discontinutiy contours, as well as a curved surface which produces a pair of extremal contours. The latter can be easily parameterized in cylindrical coordinates by an angle θ and a height z as a ruled surface:

$$\boldsymbol{X}(\theta, z) = \begin{pmatrix} a\cos\theta \\ b\sin\theta \\ 0 \end{pmatrix} + z \begin{pmatrix} ak\cos\theta \\ bk\sin\theta \\ 1 \end{pmatrix} \tag{15}$$

where a and b are minor and major half-axes of the elliptical cross-section, k is the tapering parameter of the cone and $z \in [z_1, z_2]$. It is straightforward to verify that an elliptical cone is a developable surface. Below we provide an analytical expression of its associated contour generators.

With this parametrization, equation (14) can be easily expanded to yield a trigonometric constraint of the form $F\cos\theta + G\sin\theta + H = 0$ where F, G and H depend on $\mathbf{R}(\boldsymbol{\Phi})$, $\boldsymbol{t}(\boldsymbol{\Phi})$ and C while they are independent of the parameter z. In order to solve this equation and find its roots we use the standard trigonometric substitution, i.e., $\tan\frac{\theta}{2}$ and obtain a second-degree polynomial:

$$(H - F)\tan^2\frac{\theta}{2} + 2G\tan\frac{\theta}{2} + (F + H) = 0 \tag{16}$$

This equation has two real solutions, θ_1 and θ_2, whenever the camera's optical center lies outside the cone that defines the body part (a constrained that is rarely violated). Therefore, in the case of elliptical cones the contour generator is composed of two straight lines parameterized by z, i.e., $X(\boldsymbol{\Phi}, \theta_1, z)$ and $X(\boldsymbol{\Phi}, \theta_2, z)$.

2.5 The motion of extremal contours

We turn our attention back to extremal contours – the projection onto the image plane of the contour generator. We denote by $\boldsymbol{x} = (x_1, x_2)$ the real-valued image coordinates of an extremal-contour point, i.e., eq. (7). The motion of this point depends on both:

- The *rigid motion* of the body-part with respect to the world reference frame, and
- the *sliding motion* of the contour generator onto the part's curved surface, as the relative position and orientation of this part varies with respect to the camera.

We formally derive the motion of an extremal contour point in terms of these two components. The 2-D velocity of an extremal-contour point can be predicted by:

$$\frac{d\boldsymbol{x}}{dt} = \frac{d\boldsymbol{x}}{d\boldsymbol{X}^w}\frac{d\boldsymbol{X}^w}{d\boldsymbol{r}}\frac{d\boldsymbol{r}}{d\boldsymbol{\Phi}}\frac{d\boldsymbol{\Phi}}{dt} \tag{17}$$

Vector $\boldsymbol{X}^w$, already defined by eq. (2), denotes the contour-generator point in world coordinates. Its projection is obtained from eq. (7):

$$x_1 = \frac{X_1^w}{X_3^w}\,,\, x_2 = \frac{X_2^w}{X_3^w}. \tag{18}$$

We recall that $\boldsymbol{r}$ was already defined in section 2.1 and it denotes the pose parameters associated with the body-part. Since the latter is linked to a root part by a kinematic chain, $\boldsymbol{r}$ is in it's turn parameterized by $\boldsymbol{\Phi}$. We have:

- The first term of the right-hand side of equation (17) is the image Jacobian denoted by $\mathbf{J}_I$:

$$\frac{d\boldsymbol{x}}{d\boldsymbol{X}^w} = \mathbf{J}_I = \begin{bmatrix} 1/X_3^w & 0 & -X_1^w/(X_3^w)^2 \\ 0 & 1/X_3^w & -X_2^w/(X_3^w)^2 \end{bmatrix} \tag{19}$$

- The second term is a transformation that allows to determine the velocity of a point from the motion of the part on which this point lies. When the point is rigidly attached to the part, this transformation is given by matrix $\mathbf{A}$ (see below). When the point slides onto the smooth surface there is a second transformation – matrix $\mathbf{B}$ – that remains to be determined:

$$\frac{d\boldsymbol{X}^w}{d\boldsymbol{r}} = \mathbf{A} + \mathbf{B} \tag{20}$$

- The third term is the Jacobian of the kinematic chain that links the body part to a root body part and to a world reference frame. This Jacobian matrix will be denoted by $\mathbf{J}_H$.
- The fourth term is the vector composed of both the joint velocities and the velocity of the root body part.

Now we can write eq. (17) as:

$$\dot{\boldsymbol{x}} = \mathbf{J}_I(\mathbf{A} + \mathbf{B})\mathbf{J}_H\dot{\boldsymbol{\Phi}} \tag{21}$$

where:

$$\frac{d\boldsymbol{x}}{d\boldsymbol{\Phi}} = \mathbf{J}_I(\mathbf{A} + \mathbf{B})\mathbf{J}_H \tag{22}$$

is the extremal-contour Jacobian that will be used by the tracker. It is useful to introduce the kinematic-screw notation, i.e., a six dimensional vector concatenating the rotational velocity and the translational velocity (see below):

$$\frac{d\boldsymbol{r}}{dt} = \begin{pmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V} \end{pmatrix} \tag{23}$$

The velocity of an extremal-contour point can therefore be written as:

$$\dot{\boldsymbol{x}} = \mathbf{J}_I(\mathbf{A} + \mathbf{B}) \begin{pmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V} \end{pmatrix} \tag{24}$$

Let us now make explicit the 3×6 matrices $\mathbf{A}$ and $\mathbf{B}$. By differentiation of eq. (2), we obtain:

$$\dot{\boldsymbol{X}}^w = \dot{\mathbf{R}}\boldsymbol{X} + \dot{\boldsymbol{t}} + \mathbf{R}\dot{\boldsymbol{X}} \tag{25}$$

Eq. (25) reveals that unlike the motion of a point that is rigidly attached to a surface, the motion of a contour-generator point has two components:

- A component due to the rigid motion of the smooth surface, $\dot{\mathbf{R}}\boldsymbol{X} + \dot{\boldsymbol{t}}$, and
- a component due to the sliding of the contour generator onto this smooth surface, $\mathbf{R}\dot{\boldsymbol{X}}$.

2.5.1 The rigid-motion component

The first component in eq. (25) can be parameterized by the kinematic screw and it becomes:

$$\begin{aligned} \dot{\mathbf{R}}\boldsymbol{X} + \dot{\boldsymbol{t}} &= \dot{\mathbf{R}}\mathbf{R}^\top(\boldsymbol{X}^w - \boldsymbol{t}) + \dot{\boldsymbol{t}} \\ &= \mathbf{A}\begin{pmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V} \end{pmatrix} \end{aligned} \tag{26}$$

with $[\boldsymbol{\Omega}]_\times = \dot{\mathbf{R}}\mathbf{R}^\top$, $\dot{\boldsymbol{t}} = \boldsymbol{V}$, and where $\mathbf{A}$ is the 3×6 matrix that allows to compute the velocity of a point from the kinematic screw of the rigid-body motion:

$$\mathbf{A} = [[\boldsymbol{t} - \boldsymbol{X}^w]_\times \;\; \mathbf{I}_{3\times3}] \tag{27}$$

The notation $[\boldsymbol{m}]_\times$ stands for the 3×3 skew-symmetric matrix associated with the 3-vector $\boldsymbol{m}$. Vectors $\boldsymbol{\Omega}$ and $\boldsymbol{V}$ can be concatenated to form a 6-vector $(\boldsymbol{\Omega}\ \boldsymbol{V})^\top$ which is known as the kinematic screw – the rotational and translational velocities of the body part in world coordinates.

2.5.2 The sliding-motion component

It is interesting to notice that, although the link between image contours and smooth surfaces has been thoroughly studied in the past, the problem of infering the velocity of these contours when the smooth surface undergoes a general 3-D motion has not yet been addressed. In the general case, the sliding-motion component is a complex function of both the local surface shape and of the relative motion between the surface and the observer. The problem is strongly linked to the problem of computing the aspects of a smooth surface [18] and [11] (chapters 19 & 20). Both these textbooks treat the case of a static object viewed under orthographic projection.

We establish a mathematical formalism for developable surfaces, i.e., [8] when they are viewed under perspective projection. As it has been shown above, the contour generators are rulings of the surface and their motion are fully determined by computing the time derivatives of their θ parameters. If a surface point $\boldsymbol{X}$ lies onto the contour generator, then its observed sliding velocity is:

$$\dot{\boldsymbol{X}} = \frac{\partial \boldsymbol{X}}{\partial \theta}\,\dot{\theta} + \frac{\partial \boldsymbol{X}}{\partial z}\,\dot{z} = \boldsymbol{X}_\theta\,\dot{\theta} + \boldsymbol{X}_z\,\dot{z} \tag{28}$$

The sliding velocity along the contour generator itself, $\dot{z}$, is not observable because the contour generator is the ruling of the surface – a straight line. Therefore one may assume that:

$$\dot{z} = 0 \tag{29}$$

Therefore, the sliding-motion component in eq. (25) can be written as:

$$\mathbf{R}\dot{\boldsymbol{X}} = \mathbf{R}\boldsymbol{X}_\theta\,\dot{\theta} \tag{30}$$

Since $\boldsymbol{X}$ lies onto the contour generator, it verifies the contour generator constraint, i.e., eq. (5). By differentiation of this equation we obtain a constraint for the surface parameter velocity, $\dot{\theta}$ as follows. We differentiate eq. (5), we perform the substitutions $\dot{\mathbf{R}}^\top = -\mathbf{R}^\top[\boldsymbol{\Omega}]_\times$ and $\dot{\boldsymbol{t}} = \boldsymbol{V}$, and we notice that the point velocity is tangent to the surface, i.e., $\dot{\boldsymbol{X}}^\top\boldsymbol{n} = 0$. We obtain an expression for the derivative of eq. (5):

$$\left(\boldsymbol{X} + \mathbf{R}^\top(\boldsymbol{t} - \boldsymbol{C})\right)^\top \dot{\boldsymbol{n}} = \left([\boldsymbol{\Omega}]_\times(\boldsymbol{t} - \boldsymbol{C}) - \boldsymbol{V}\right)^\top \mathbf{R}\boldsymbol{n} \tag{31}$$

With $\dot{\boldsymbol{n}} = \boldsymbol{n}_\theta\,\dot{\theta}$ and with $[\boldsymbol{a}]_\times\boldsymbol{b} = -[\boldsymbol{b}]_\times\boldsymbol{a}$, we obtain from eq. (31):

$$\dot{\theta} = \frac{(\mathbf{R}\boldsymbol{n})^\top\,[[\boldsymbol{C} - \boldsymbol{t}]_\times \quad -\mathbf{I}_{3\times 3}]}{(\boldsymbol{X} + \mathbf{R}^\top(\boldsymbol{t} - \boldsymbol{C}))^\top\boldsymbol{n}_\theta} \begin{pmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V} \end{pmatrix} \tag{32}$$

Therefore, the sliding velocity $\dot{\theta}$ can be expressed as a function of (i) the surface parameterization, (ii) the relative position and orientation of the camera with respect to the surface, and (iii) the rigid motion of the surface (the kinematic screw). To summarize, eq. (30) becomes:

$$\mathbf{R}\dot{\boldsymbol{X}} = \mathbf{B}\begin{pmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V} \end{pmatrix} \tag{33}$$

where $\mathbf{B}$ is the 3×6 matrix:

$$\mathbf{B} = \frac{1}{b}\,\mathbf{R}\boldsymbol{X}_\theta\,(\mathbf{R}\boldsymbol{n})^\top\,[[\boldsymbol{C} - \boldsymbol{t}]_\times \quad -\mathbf{I}_{3\times 3}]$$

and the scalar b is defined by:

$$b = (\boldsymbol{X} + \mathbf{R}^\top(\boldsymbol{t} - \boldsymbol{C}))^\top\boldsymbol{n}_\theta$$

2.5.3 The velocity of extremal contours

To conclude this section, the velocity of an extremal contour point has a rigid-motion component and a surface-sliding component:

$$\dot{\boldsymbol{x}} = \dot{\boldsymbol{x}}^r + \dot{\boldsymbol{x}}^s \tag{34}$$

The explicit parameterization of the sliding component, as shown above, allows its incorporation into the explicit representation of the observed image velocities as a function of the kinematic-chain parameters, as described in detail below.

The sliding velocity depends both on the curvature of the surface and on the velocity of the surface. In practice it will speed up the convergence of the tracker by a factor of two, as described in section 6.

3 The human-body kinematic chain

In the case of a kinematic chain, the rigid motion of a body-part can be parameterized by the joint parameters. Kinematic chains are widely used by human-body trackers and motion capture systems. In this section we introduce the use of the *zero-reference kinematic representation* for modelling the human-body articulated chains. The Zero-reference kinematic representation was studied for robot manipulators [21] and [20]. The parameterization introduced in this section combines the zero-reference representation with the free motion of the root body-part, i.e., Figure 3.

Without loss of generality we consider any one among the several kinematic chains needed to describe the human body. A body part B is linked to the root body-part R by a kinematic chain with p rotational degrees of freedom. The root body part itself moves freely with six degrees of freedom (three rotations and three translations) and with respect to the world coordinate frame. Let $\boldsymbol{\Lambda} = (\lambda_1, \ldots, \lambda_p)$ denote the joint angles associated with the kinematic chain, and let $\boldsymbol{\Psi} = (\psi_1, \ldots, \psi_q)$ denote the rotational and translational degrees of freedom of the free motion. In the most general case we have $q = 6$. Therefore, there are $p + q$ motion parameters embeded in the vector $\boldsymbol{\Phi} = (\boldsymbol{\Psi}, \boldsymbol{\Lambda})$.

With the same notations as in the previous section, we consider a point X that belongs to the contour generator associated with a developable surface and body part. The point's *homogeneous coordinates* in the local frame are denoted by $\overline{\boldsymbol{X}} = (X_1 \ X_2 \ X_3 \ 1)^\top$. We also denote by $\boldsymbol{X}^r$ the coordinates of the same point in the root body-part frame, and by $\boldsymbol{X}^w$ its coordinates in the world frame.

Moreover, we denote with $\mathbf{D}(\boldsymbol{\Psi})$ the 4×4 homogeneous matrix associated with the free motion of the root body part with respect to a fixed world frame, and with $\mathbf{K}(\boldsymbol{\Lambda})$ the 4×4 homogeneous matrix associated with the constrained motion of a body part with respect to the root part. Let $\boldsymbol{\Lambda}^0$ be the joint angles for a particular *reference position* of the kinematic chain. Obviously we have $\overline{\boldsymbol{X}}^r(\boldsymbol{\Lambda}) = \mathbf{K}(\boldsymbol{\Lambda})\overline{\boldsymbol{X}}$ and $\overline{\boldsymbol{X}}^r(\boldsymbol{\Lambda}^0) = \mathbf{K}(\boldsymbol{\Lambda}^0)\overline{\boldsymbol{X}}$. We obtain $\overline{\boldsymbol{X}}^r(\boldsymbol{\Lambda}) = \mathbf{K}(\boldsymbol{\Lambda})\mathbf{K}^{-1}(\boldsymbol{\Lambda}^0)\overline{\boldsymbol{X}}^r(\boldsymbol{\Lambda}^0)$. With this formula and from $\overline{\boldsymbol{X}}^w(\boldsymbol{\Psi}, \boldsymbol{\Lambda}) = \mathbf{D}(\boldsymbol{\Psi})\overline{\boldsymbol{X}}^r(\boldsymbol{\Lambda})$ we obtain:

$$\overline{\boldsymbol{X}}^w(\boldsymbol{\Psi}, \boldsymbol{\Lambda}) = \mathbf{D}(\boldsymbol{\Psi})\mathbf{K}(\boldsymbol{\Lambda})\mathbf{K}^{-1}(\boldsymbol{\Lambda}^0)\overline{\boldsymbol{X}}^r(\boldsymbol{\Lambda}^0)$$

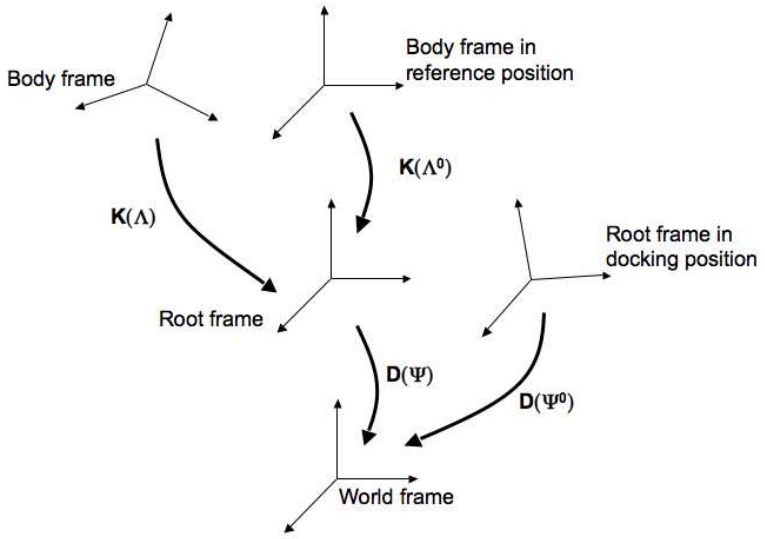


Figure 3: Each body part has a frame associated with it, therefore motions are represented by changes in coordinate frames. There is a reference position for each body part in the chain defined by the joint angles $\mathbf{\Lambda}^0$. Similarly, there is a docking position for the root body-part defined by the six-dimensional vector $\mathbf{\Psi}^0$.

We also consider a reference or a *docking* position for the root body-part, defined by the free-motion paramaters $\mathbf{\Psi}^0$, i.e., $\overline{\boldsymbol{X}}^w(\mathbf{\Psi}^0, \mathbf{\Lambda}^0) = \mathbf{D}(\mathbf{\Psi}^0)\overline{\boldsymbol{X}}^r(\mathbf{\Lambda}^0)$. Finally we obtain:

$$\begin{aligned} \overline{\boldsymbol{X}}^w(\mathbf{\Psi}, \mathbf{\Lambda}) &= \mathbf{D}(\mathbf{\Psi})\mathbf{K}(\mathbf{\Lambda})\mathbf{K}^{-1}(\mathbf{\Lambda}^0)\mathbf{D}^{-1}(\mathbf{\Psi}^0)\overline{\boldsymbol{X}}^w(\mathbf{\Psi}^0, \mathbf{\Lambda}^0) && (35)\\ &= \mathbf{H}(\mathbf{\Psi}, \mathbf{\Psi}^0, \mathbf{\Lambda}, \mathbf{\Lambda}^0)\overline{\boldsymbol{X}}^w(\mathbf{\Psi}^0, \mathbf{\Lambda}^0) && (36)\end{aligned}$$

It will be convenient to write the above transformation as:

$$\mathbf{H}(\mathbf{\Psi}, \mathbf{\Psi}^0, \mathbf{\Lambda}, \mathbf{\Lambda}^0) = \mathbf{F}(\mathbf{\Psi}, \mathbf{\Psi}^0)\ \mathbf{Q}(\mathbf{\Lambda}, \mathbf{\Lambda}^0, \mathbf{\Psi}^0) \tag{37}$$

with:

$$\mathbf{F}(\mathbf{\Psi}, \mathbf{\Psi}^0) = \mathbf{D}(\mathbf{\Psi})\mathbf{D}^{-1}(\mathbf{\Psi}^0) \tag{38}$$

and:

$$\mathbf{Q}(\mathbf{\Lambda}, \mathbf{\Lambda}^0, \mathbf{\Psi}^0) = \mathbf{D}(\mathbf{\Psi}^0)\mathbf{K}(\mathbf{\Lambda})\mathbf{K}^{-1}(\mathbf{\Lambda}^0)\mathbf{D}^{-1}(\mathbf{\Psi}^0) \tag{39}$$

3.1 The kinematic-chain model

The transformation **K** describes an open kinematic chain and the transformation **Q** describes exactly the same chain but relatively to a *reference position* of the chain. **K** may be written as a composition of fixed transformations $\mathbf{L}_1, \ldots, \mathbf{L}_p$, and of one-degree-of-freedom rotations $\mathbf{J}(\lambda_1), \ldots, \mathbf{J}(\lambda_p)$:

$$\mathbf{K}(\mathbf{\Lambda}) = \mathbf{L}_1\mathbf{J}(\lambda_1)\ldots\mathbf{L}_p\mathbf{J}(\lambda_p) \tag{40}$$

where the matrices $\mathbf{L}_1 \ldots \mathbf{L}_p$ are fixed transformations between adjacent rotational joints, and matrices of the form of **J** are the canonical representations of a rotation, i.e., the z-axis is aligned with the rotation axis:

$$\mathbf{J}(\lambda) = \begin{bmatrix} \cos\lambda & -\sin\lambda & 0 & 0 \\ \sin\lambda & \cos\lambda & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Notice that $\mathbf{J}(\lambda)$ describes a commutative group. Moreover, it is a one-dimensional Lie group parameterized by the angle λ. The inverse of this matrix is $\mathbf{J}(-\lambda)$. The representation outlined above allows one to describe rotational joints with one, two, or three degrees of freedom as a kinematic chain as well. For example, a rotational joint $\mathbf{J}_{\alpha,\beta,\gamma}$ with three degrees of freedom can be parameterized by three Euler angles, α, β, γ, and can be written as $\mathbf{J}(\alpha)\mathbf{L}\mathbf{J}(\beta)\mathbf{L}\mathbf{J}(\gamma)\mathbf{L}$. In this particular case, the fixed transformation has the form:

$$\mathbf{L} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matrix **Q** in eq. (39) can now be written as a product of one-degree-of-freedom transformations $\mathbf{Q}_i$:

$$\mathbf{Q}(\mathbf{\Lambda}, \mathbf{\Lambda}^0, \mathbf{\Psi}^0) = \mathbf{Q}_1(\lambda_1 - \lambda_1^0)\ldots\mathbf{Q}_i(\lambda_i - \lambda_i^0)\ldots\mathbf{Q}_p(\lambda_p - \lambda_p^0) \tag{41}$$

where each term $\mathbf{Q}_i$ is of the form, i.e., [20]:

$$\mathbf{Q}_i(\lambda_i - \lambda_i^0) = \underbrace{\mathbf{D}(\mathbf{\Psi}^0)\mathbf{L}_1\mathbf{J}(\lambda_1^0)\ldots\mathbf{L}_i}_{\mathbf{U}_i}\ \mathbf{J}(\lambda_i - \lambda_i^0)\ \underbrace{\mathbf{L}_i^{-1}\ldots\mathbf{J}(-\lambda_1^0)\mathbf{L}_1^{-1}\mathbf{D}^{-1}(\mathbf{\Psi}^0)}_{\mathbf{U}_i^{-1}} \tag{42}$$

Therefore, the transformation $\mathbf{Q}_i(\lambda_i - \lambda_i^0)$ is conjugated to $\mathbf{J}(\lambda_i - \lambda_i^0)$:

$$\mathbf{Q}_i = \mathbf{U}_i\mathbf{J}\mathbf{U}_i^{-1}$$

Notice that matrices $\mathbf{U}_i, \{i = 1 \ldots p\}$ remain fixed when the joint parameters vary **and** when the root body-part undergoes a free motion. Each matrix $\mathbf{Q}_i$ in eq. (41) may be interpreted as a one-dimensional Lie Group. Others used the exponential representation for this one-dimensional transformations, [22], [4].

3.2 The zero-reference kinematic model

Without loss of generality, one may set the initial joint-angle values to zero, i.e., $\lambda_1^0 = \ldots = \lambda_p^0 = 0$. In this case, the kinematic chain does not depend any more on its reference pose, since $\mathbf{J}(\lambda_i^0) = \mathbf{I}$ for all i. The kinematic chain writes in this case:

$$\mathbf{Q}(\mathbf{\Lambda}, \mathbf{\Psi}^0) = \mathbf{Q}_1(\lambda_1, \mathbf{\Psi}^0) \ldots \mathbf{Q}_i(\lambda_i, \mathbf{\Psi}^0) \ldots \mathbf{Q}_p(\lambda_p, \mathbf{\Psi}^0) \tag{43}$$

The human-body zero-reference kinematic chain. From the equations above, one may write a compact and convenient factorization of matrix $\mathbf{H}$, i.e., eq. (37):

$$\mathbf{H}(\mathbf{\Psi}, \mathbf{\Psi}^0, \mathbf{\Lambda}) = \mathbf{F}(\mathbf{\Psi}, \mathbf{\Psi}^0)\mathbf{Q}_1(\lambda_1, \mathbf{\Psi}^0) \ldots \mathbf{Q}_i(\lambda_i, \mathbf{\Psi}^0) \ldots \mathbf{Q}_p(\lambda_p, \mathbf{\Psi}^0) \tag{44}$$

4 The Jacobian of the human-body kinematic chain

In this section we make explicit the Jacobian matrix associated with the kinematic chain of the human-body, $\mathbf{J}_H$. This matrix appears in eq. (17); From this equation and from eq. (24) we obtain:

$$\begin{pmatrix} \mathbf{\Omega} \\ \boldsymbol{V} \end{pmatrix} = \mathbf{J}_H \dot{\mathbf{\Phi}} \tag{45}$$

The Jacobian of a kinematic chain such as the one described above is intrinsic to the mechanical and geometric structure of the kinematic chain and it does not depend on a particular choice of a point X, is it sliding onto the surface or rigidly attached to it. The Jacobian $\mathbf{J}_H$ maps joint velocities onto the kinematic screw of a body part whose kinematic chain is denoted by $\mathbf{H}$. In order to establish an expression for the Jacobian, we will first need to determine the tangent operator $\widehat{\mathbf{H}}$ of $\mathbf{H}$:

$$\widehat{\mathbf{H}} = \dot{\mathbf{H}}\mathbf{H}^{-1}, \tag{46}$$

and second, parameterize $\widehat{\mathbf{H}}$ such that it depends only on the kinematic parameters, i.e., we must take the derivative of a body-part point with respect to the motion variables, i.e., $d\boldsymbol{X}^w/d\mathbf{\Phi}$. The case of human-body motion is different than the classical case studied in the robotics literature, [20], [21], [22] because one must take into account the fact that the root-part of the chain undergoes a free rigid motion.

4.1 A rotational joint

First, we consider the case of a single rotational joint. It's tangent operator is defined by $\widehat{\mathbf{Q}}_i = \dot{\mathbf{Q}}_i\mathbf{Q}_i^{-1}$, and we obviously have $\widehat{\mathbf{Q}}_i = \mathbf{U}_i\widehat{\mathbf{J}}_i\mathbf{U}_i^{-1}$. From the expression of $\mathbf{J}$ we have:

$$\widehat{\mathbf{J}}(\lambda_i - \lambda_i^0) = \widehat{\mathbf{J}}(\lambda_i) = \dot{\lambda}_i \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \dot{\lambda}_i \widetilde{\mathbf{J}}$$

Therefore, we obtain a simple expression for the tangent operator associated with one joint:

$$\widehat{\mathbf{Q}}_i(\lambda_i - \lambda_i^0) = \dot{\lambda}_i \mathbf{U}_i \widetilde{\mathbf{J}} \mathbf{U}_i^{-1} = \dot{\lambda}_i \widetilde{\mathbf{Q}}_i \tag{47}$$

Matrix $\widetilde{\mathbf{J}}$ is called the *Lie-algebra of the Lie-group* defined by the matrices of the form of $\mathbf{J}$. If one prefers the exponential representation, $\widetilde{\mathbf{J}}$ is called a *twist*. Notice that the upper 3×3 block of this matrix is skew-symmetric. Matrix $\widetilde{\mathbf{Q}}_i = \mathbf{U}_i \widetilde{\mathbf{J}} \mathbf{U}_i^{-1}$ is the Lie-algebra of $\mathbf{Q}$ which spans the same Lie group as $\mathbf{J}$ but in a different reference frame. Since $\widetilde{\mathbf{J}}$ has a skew-symmetric 3×3 upper-left block, it is easy to verify that $\mathbf{U}\widetilde{\mathbf{J}}\mathbf{U}^{-1}$ has the same structure for any rigid transformation $\mathbf{U}$.

4.2 The tangent operator

Second, we determine the tangent operator of the human-body kinematic chain. The zero-reference kinematic chain – $\mathbf{H}$ is composed of a rotation matrix and a translation vector: $\mathbf{R}_H$, $\boldsymbol{t}_H$. We derive a formula for the velocity of a point lying onto the contour generator of a body-part, i.e., eqs. (26), (33):

$$\begin{aligned}
\dot{\boldsymbol{X}}^w(\boldsymbol{\Psi}, \boldsymbol{\Lambda}) &= \dot{\mathbf{R}}_H \boldsymbol{X}^w(\boldsymbol{\Psi}_0, \boldsymbol{\Lambda}_0) + \dot{\boldsymbol{t}}_H + \mathbf{R}_H \dot{\boldsymbol{X}}^w(\boldsymbol{\Psi}_0, \boldsymbol{\Lambda}_0) \\
&= \dot{\mathbf{R}}_H \mathbf{R}_H^\top (\boldsymbol{X}^w(\boldsymbol{\Psi}, \boldsymbol{\Lambda}) - \boldsymbol{t}_H) + \dot{\boldsymbol{t}}_H + \mathbf{R}_H \dot{\boldsymbol{X}}^w(\boldsymbol{\Psi}_0, \boldsymbol{\Lambda}_0) \\
&= (\mathbf{A}_H + \mathbf{B}_H) \begin{pmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V} \end{pmatrix}
\end{aligned} \tag{48}$$

We remind that the velocity of a point lying onto the contour generator has two components, the rigid motion component $\mathbf{A}_H$ and the sliding motion one $\mathbf{B}_H$. Vectors $\boldsymbol{\Omega}$ and $\boldsymbol{V}$ form the kinematic screw for which we seek an expression:

$$\widehat{\mathbf{H}}(\boldsymbol{\Psi}, \boldsymbol{\Lambda}) = \begin{bmatrix} [\boldsymbol{\Omega}]_\times & \boldsymbol{V} \\ \mathbf{0}^\top & 0 \end{bmatrix} \tag{49}$$

Since $\mathbf{H} = \mathbf{F}\mathbf{Q}$, we have $\dot{\mathbf{H}} = \dot{\mathbf{F}}\mathbf{Q} + \mathbf{F}\dot{\mathbf{Q}}$ and:

$$\widehat{\mathbf{H}}(\boldsymbol{\Psi}, \boldsymbol{\Lambda}) = \widehat{\mathbf{F}}(\boldsymbol{\Psi}) + \mathbf{F}\widehat{\mathbf{Q}}(\boldsymbol{\Lambda})\mathbf{F}^{-1} \tag{50}$$

As detailed below, the tangent operator can be written as the sum:

$$\widehat{\mathbf{H}}(\boldsymbol{\Psi}, \boldsymbol{\Lambda}) = \widehat{\mathbf{H}}_r(\boldsymbol{\Psi}) + \sum_{i=1}^{p} \widehat{\mathbf{H}}_i(\lambda_i) \tag{51}$$

- *The tangent operator associated with the free motion of the root body-part,* $\widehat{\mathbf{H}}_r(\boldsymbol{\Psi}) = \widehat{\mathbf{F}}(\boldsymbol{\Psi})$; From eq. (38) we obtain: $\widehat{\mathbf{F}}(\boldsymbol{\Psi}) = \widehat{\mathbf{D}}(\boldsymbol{\Psi})$. $\widehat{\mathbf{H}}_r$ is the 4×4 matrix parameterized by the rotational velocity $\boldsymbol{\omega}_r$ and the translational velocity $\boldsymbol{\nu}_r$ of this free motion:

$$\widehat{\mathbf{H}}_r(\boldsymbol{\Psi}) = \begin{bmatrix} [\boldsymbol{\omega}_r]_\times & \boldsymbol{\nu}_r \\ \mathbf{0}^\top & 0 \end{bmatrix} \tag{52}$$

This motion has six degrees of freedom and can be parameterized by three rotations and three translations:

$$\mathbf{R} = \mathbf{R}_z(\psi_3)\mathbf{R}_y(\psi_2)\mathbf{R}_x(\psi_1)$$
$$\boldsymbol{t} = \psi_4\boldsymbol{e}_x + \psi_5\boldsymbol{e}_y + \psi_6\boldsymbol{e}_z,$$

where $\boldsymbol{e}_x = (1\ 0\ 0)^\top$ and so forth. The kinematic screw of this motion can therefore be written as:

$$\begin{pmatrix} \boldsymbol{\omega}_r \\ \boldsymbol{\nu}_r \end{pmatrix} = \begin{bmatrix} \boldsymbol{\omega}_x & \boldsymbol{\omega}_y & \boldsymbol{\omega}_z & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{e}_x & \boldsymbol{e}_y & \boldsymbol{e}_z \end{bmatrix}_{6\times 6} \dot{\boldsymbol{\Psi}} \tag{53}$$

with $[\boldsymbol{\omega}_z]_\times = [\boldsymbol{e}_z]_\times$, $[\boldsymbol{\omega}_y]_\times = \mathbf{R}_z[\boldsymbol{e}_y]_\times\mathbf{R}_z^\top$, and $[\boldsymbol{\omega}_x]_\times = \mathbf{R}_y\mathbf{R}_z[\boldsymbol{e}_x]_\times\mathbf{R}_z^\top\mathbf{R}_y^\top$.

- *The tangent operator associated with the constrained motion of the kinematic chain,* $\mathbf{F}\widehat{\mathbf{Q}}(\boldsymbol{\Lambda})\mathbf{F}^{-1}$; it is expressed in world coordinates and with respect to a reference position defined by both $\boldsymbol{\Psi}^0$ and $\boldsymbol{\Lambda}^0$. This tangent operator can be expanded as:

$$\begin{aligned} \widehat{\mathbf{Q}}(\boldsymbol{\Lambda}) &= \widehat{\mathbf{Q}}_1(\lambda_1) + \mathbf{Q}_1(\lambda_1)\widehat{\mathbf{Q}}_2(\lambda_2)\mathbf{Q}_1^{-1}(\lambda_1) + \ldots + \\ &\quad \mathbf{Q}_1(\lambda_1)\ldots\mathbf{Q}_{p-1}(\lambda_{p-1})\ \widehat{\mathbf{Q}}_p(\lambda_p)\ \mathbf{Q}_{p-1}^{-1}(\lambda_{p-1})\ldots\mathbf{Q}_1^{-1}(\lambda_1) \\ &= \dot{\lambda}_1\widetilde{\mathbf{Q}}_1 + \dot{\lambda}_2\mathbf{Q}_1\widetilde{\mathbf{Q}}_2\mathbf{Q}_1^{-1} + \ldots + \dot{\lambda}_p\mathbf{Q}_1\ldots\mathbf{Q}_{p-1}\ \widetilde{\mathbf{Q}}_p\ \mathbf{Q}_{p-1}^{-1}\ldots\mathbf{Q}_1^{-1} \end{aligned} \tag{54}$$

Therefore, the tangent operator associated with the i-th link of the kinematic-chain is:

$$\widehat{\mathbf{H}}_i = \dot{\lambda}_i\ \mathbf{F}(\boldsymbol{\Psi})\mathbf{Q}_1(\lambda_1)\ldots\mathbf{Q}_{i-1}(\lambda_{i-1})\mathbf{U}_i\widetilde{\mathbf{J}}\mathbf{U}_i^{-1}\mathbf{Q}_{i-1}^{-1}(\lambda_{i-1})\ldots\mathbf{Q}_1^{-1}(\lambda_1)\mathbf{F}^{-1}(\boldsymbol{\Psi}) \tag{55}$$

Obviously, this matrix simply writes:

$$\widehat{\mathbf{H}}_i(\boldsymbol{\Psi}, \lambda_1, \ldots, \lambda_{i-1}) = \dot{\lambda}_i\widetilde{\mathbf{H}}_i = \dot{\lambda}_i \begin{bmatrix} [\boldsymbol{\omega}_i]_\times & \boldsymbol{\nu}_i \\ \mathbf{0}^\top & 0 \end{bmatrix} \tag{56}$$

Therefore, by combining eqs. (51),(53), and (53) we obtain the following expression for the kinematic screw:

$$\begin{pmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V} \end{pmatrix} = \begin{bmatrix} \boldsymbol{\omega}_x & \boldsymbol{\omega}_y & \boldsymbol{\omega}_z & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{e}_x & \boldsymbol{e}_y & \boldsymbol{e}_z \end{bmatrix} \dot{\boldsymbol{\Psi}} + \begin{bmatrix} \boldsymbol{\omega}_1 & \ldots & \boldsymbol{\omega}_p \\ \boldsymbol{\nu}_1 & \ldots & \boldsymbol{\nu}_p \end{bmatrix} \dot{\boldsymbol{\Lambda}} \tag{57}$$

with $\dot{\boldsymbol{\Psi}} = (\dot{\psi}_1\ldots\dot{\psi}_6)^\top$ and $\dot{\boldsymbol{\Lambda}} = (\dot{\lambda}_1\ldots\dot{\lambda}_p)^\top$. Finally, the Jacobian of the human-body kinematic chain writes as a $6\times(6+p)$ matrix:

$$\mathbf{J}_H = \begin{bmatrix} \boldsymbol{\omega}_x & \boldsymbol{\omega}_y & \boldsymbol{\omega}_z & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\omega}_1 & \ldots & \boldsymbol{\omega}_p \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{e}_x & \boldsymbol{e}_y & \boldsymbol{e}_z & \boldsymbol{\nu}_1 & \ldots & \boldsymbol{\nu}_p \end{bmatrix} \tag{58}$$

5 Fitting extremal contours to image contours

In this section we consider the problem of fitting extremal contours – model contours predicted in the image plane, with image contours – contours extracted from the data. Therefore, we have to measure the discrepancy between a set of predictions (extremal contours) and a set of observations (image contours): we want to find the model's parameters that minimize this discrepancy.

Although the human body comprises several (five) kinematic chains, for the sake of clarity of exposition we consider only one such kinematic chain. We collect extremal-contour points from all the body-parts. Let $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_j, \ldots, \boldsymbol{x}_m\}$ be the prediction vector, a set of m extremal-contour points. The components of this vector are 2-D points and they are parameterized by the kinematic- and free-motion parameter vector $\boldsymbol{\Phi}$, i.e. $\boldsymbol{x}(\boldsymbol{\Phi})$. Similarly, let $\mathcal{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_i, \ldots, \boldsymbol{y}_k\}$ be the observation vector – a set of contour points observed in the image. In order to estimate the motion parameters one has to compare these two sets through a metric and to minimize it over the motion variables. Therefore, the problem can be generally stated as the minimization of a multi-variate scalar function E of eq. (1).

There are several ways of defining and measuring the distance between two sets of points, $\mathcal{Y}$ and $\mathcal{X}$. One way of measuring this distance is to sum over one-to-one pairings $(\boldsymbol{x}_j, \boldsymbol{y}_i)$:

$$E(\mathcal{Y}, \mathcal{X}(\boldsymbol{\Phi})) = \sum_i \sum_j \alpha_{ij} \|\boldsymbol{y}_i - \boldsymbol{x}_j(\boldsymbol{\Phi})\|^2 \tag{59}$$

where the *hidden* α_{ij} variables are the entries of an association matrix: $\alpha_{ij} = 1$ if the observable $\boldsymbol{y}_i$ matches the prediction $\boldsymbol{x}_j$, and $\alpha_{ij} = 0$ otherwise. Therefore one has to solve both for the hidden variables and for the motion parameters [5].

5.1 The Hausdorf distance

Another way of measuring the distance between two point-sets is to use the *Hausdorff distance* [15, 23] which does not make use of explicit point pairings:

$$H(\mathcal{Y}, \mathcal{X}) = \max(h(\mathcal{Y}, \mathcal{X}), h(\mathcal{X}, \mathcal{Y})) \tag{60}$$

where $h()$ is called the *directed* Hausdorff distance:

$$h(\mathcal{Y}, \mathcal{X}) = \max_i(\min_j(\|\boldsymbol{y}_i - \boldsymbol{x}_j\|))$$

The function $h()$ identifies the point in $\mathcal{Y}$ which is the farthest from any point in $\mathcal{X}$. The Hausdorff distance is the maximum of $h(\mathcal{Y}, \mathcal{X})$ and of $h(\mathcal{X}, \mathcal{Y})$ and hence it measures the degree of mismatch between two point sets *without making explicit pairings* of points in one set with points in the other set. This means that many points of $\mathcal{Y}$ may be assigned to the same point of $\mathcal{X}$.

5.2 The chamfer distance

If the $\max$ operator in the Hausdorff distance is replaced by the summation operator, we obtain the normalized *directed* (or non-symmetric) chamfer distance:

$$DCD(\mathcal{Y},\mathcal{X}) = \frac{1}{k}\sum_{i=1}^{k}\min_{j}(\|\boldsymbol{y}_i - \boldsymbol{x}_j\|) \tag{61}$$

The directed chamfer distance, or DCD, is a positive function and has the properties of identity and of triangle inequality but not of symmetry. It also has the desirable property that it can be computed very efficiently. Indeed, the DCD can be computed from the binary image of the observed image contour set $\mathcal{Y}$ using the *chamfer-distance image* $C_{\mathcal{Y}}$ [3, 13]. The subscript reminds that this image is associated with the set $\mathcal{Y}$ of observed edge points. For each image site (pixel) with integer-valued image coordinates u_1 and u_2, the chamfer-distance image $C_{\mathcal{Y}}(u_1,u_2)$ returns the real-valued distance from this pixel to the nearest contour point of $\mathcal{Y}$. Therefore one can evalute the distance from a predicted extremal-contour point $\boldsymbol{x} \in \mathcal{X}$ to its closest image contour by evaluating the chamfer-distance image at $\boldsymbol{x}$ with real-valued image coordinates x_1 and x_2.

We denote by $[x]$ the integer part of a real number x. Let $u_1 = [x_1]$ and $u_2 = [x_2]$ be the integer parts, and $r_1 = x_1 - [x_1]$ and $r_2 = x_2 - [x_2]$ be the fractional parts of the coordinates of a predicted point $\boldsymbol{x}$. The chamfer distance at $\boldsymbol{x}$ can be obtained by bi-linear interpolation of the chamfer-distance image:

$$\begin{aligned} D(\mathcal{Y},\boldsymbol{x}) \quad = \quad & (1-r_1)(1-r_2)C_{\mathcal{Y}}(u_1,u_2) + r_1(1-r_2)C_{\mathcal{Y}}(u_1+1,u_2) \\ & +(1-r_1)r_2C_{\mathcal{Y}}(u_1,u_2+1) + r_1r_2C_{\mathcal{Y}}(u_1+1,u_2+1) \end{aligned} \tag{62}$$

5.3 Minimizing the chamfer distance

The minimization problem defined by eq. (1) can be rewritten as the sum of squares of the chamfer distances over the predicted model contours:

$$f(\boldsymbol{\Phi}) = \frac{1}{2}\sum_{j=1}^{m} D_j^2(\mathcal{Y},\boldsymbol{x}_j(\boldsymbol{\Phi})) = \frac{1}{2}\sum_{j=1}^{m} D_j^2(\boldsymbol{\Phi}) \tag{63}$$

In order to minimize this function over the motion parameters, we take its second-order Taylor expansion as well as the Gauss-Newton approximation of the Hessian:

$$f(\boldsymbol{\Phi}+\boldsymbol{d}) = f(\boldsymbol{\Phi}) + \boldsymbol{d}^\top \mathbf{J}_D^\top \boldsymbol{D} + \frac{1}{2}\boldsymbol{d}^\top \mathbf{J}_D^\top \mathbf{J}_D \boldsymbol{d} + \dots$$

where $\boldsymbol{D}^\top = \begin{pmatrix} D_1 & \dots & D_m \end{pmatrix}$ and $\mathbf{J}_D^\top = [d\boldsymbol{D}/d\boldsymbol{\Phi}]^\top$ is the $n \times m$ matrix:

$$\mathbf{J}_D^\top = \begin{bmatrix} \frac{d\,D_1}{d\,\boldsymbol{\Phi}} & \dots & \frac{d\,D_m}{d\,\boldsymbol{\Phi}} \end{bmatrix} \tag{64}$$

The chamfer-distance gradient. The embedding of the tracker into such an optimization framework requires an analytic expression for the gradient of the error function to be minimized. The derivative of the chamfer distance D_j with respect to the motion parameters is the following matrix product:

$$\frac{d\,D_j}{d\,\mathbf{\Phi}} = \left(\frac{d\,D_j}{d\,\boldsymbol{x}}\right)^{\top} \frac{d\,\boldsymbol{x}}{d\,\mathbf{\Phi}}$$

By noticing that $d[x]/dx = 0$, we immediately obtain an expression for $dD_j/d\boldsymbol{x}$:

$$\begin{aligned}
\frac{\partial D_j}{d\,x_1} &= (1-r_2)(C_{\mathcal{Y}}(u_1+1,u_2) - C_{\mathcal{Y}}(u_1,u_2)) \\
&\quad + r_2(C_{\mathcal{Y}}(u_1+1,u_2+1) - C_{\mathcal{Y}}(u_1,u_2+1)) \\
\frac{\partial D_j}{d\,x_2} &= (r_1-1)(C_{\mathcal{Y}}(u_1+1,u_2) + C_{\mathcal{Y}}(u_1,u_2)) \\
&\quad + r_1(C_{\mathcal{Y}}(u_1+1,u_2+1) + C_{\mathcal{Y}}(u_1,u_2+1))
\end{aligned}$$

We remind that $d\boldsymbol{x}/d\mathbf{\Phi} = \mathbf{J}_I(\mathbf{A}+\mathbf{B})\mathbf{J}_H$ is the extremal-contour Jacobian defined in eq. (22).

The practical issues related to the minimization of the chamfer distance can be found in [17]. Here we analyse the practical conditions under which this minimization should be carried out. At each time instant, the tracker is initialized with the previously found solution and eq. (63) must be minimized. This minimization problem needs one necessary condition, namely that the $n \times n$ Hessian matrix has full rank. The Jacobian $\mathbf{J}_D$ is of size $m \times n$ and we recall that n is the number of variables to be estimated (the motion parameters) and m is the number of predictions (extremal contour points). To compute the inverse of $\mathbf{J}_D^{\top}\mathbf{J}_D$ we must have $m \geq n$ with n independent matrix rows.

Since each prediction accounts for one row in the Jacobian matrix, one must somehow insure that there are n "independent" predictions. If each body part is viewed as a rigid object in motion, then it has six degrees of freedom. A set of three non-colinear points constrain these degrees of freedom. Whenever there are one-to-one model-point-to-image-point assignments, a set of three points is sufficient to constrain all six degrees of freedom. In the case of the chamfer distance there are no such one-to-one assignments and each model point yields only one constrain. Therefore, when one uses the chamfer distance, the problem is unconstrained since three non-colinear points yield three constraints only. Within a kinematic chain, the root body-part has six degrees of freedom and each body-part has 6+p degrees of freedom. Fortunately the body-parts are linked together to form kinematic chains. Therefore, one sensible hypothesis is to assume that the points at hand are evenly distributed among the body parts.

The kinematic human-body model that we use is composed of 5 kinematic chains that share a common root body-part, 19 body-parts, and 54 degrees of freedom (48 rotational joints and 6 free-motion parameters). Therefore, with an average of 3 points per body-part, there are in principle enough constraints to solve the tracking problem. Notice that the root-body part can arbitrarily be

chosen and there is no evidence that one body-part is more suitable than another body-part to be the root part. This flexibility can be exploited in practice as described in the next section.

In practice there are other difficulties and problems. Due to total and/or partial occlusions, not all the body-parts can be predicted visible in one image. Therefore, it is impossible to insure that that all the degrees of freedom are actually measured in one image. Even if a point attached to a visible body-part is predicted in the image, it may not be present in the data and/or it may be badly extracted and located. Non-relevant edges that lie in the neighbourhood of a predicted location contribute to the chamfer distance and therefore complicate the task of the minimization process.

One way to increase the robustness of the tracker it to make recourse to redundant data. The latter may be obtained by using several cameras, each camera providing an independent chamfer distance error function. Provided that the cameras are *calibrated and synchronized* the method described above can be simultenously applied to all the cameras. There will be several Jacobian matrices of the form of eq. (64) (one for each camera) and these matrices can be combined together in a unique Jacobian, provided that a common world reference frame is being used [19]. Therefore, one increases the number of predictions (columns in the Jacobian) without increasing the number of variables.

It is worthwhile to notice that the extremal contours viewed with one camera are different than the extremal contours viewed with another camera. Indeed, these two sets of contours correspond to different physical points onto the surface. One great advantage of this feature is that there is no need to establish point-to-point matches between images taken with distinct cameras.

6 Experimental results

The method described in this report was successfully applied to both simulated and real data.

6.1 Tracking simulated motions

First, the simulated data was produced using a motion capture system and an animation software. This simulation outputs trajectories for the motion parameters of our human-body model. From these trajectories we generated a sequence of model motions. From each one of the model poses we computed and projected the corresponding silhouettes onto six image planes associated with six virtual cameras. There were 120 images per video.

Next, we applied our method to these silhouettes. Figure 4 shows the simulated poses (top rows) as well as the estimated poses (bottom rows). Figures 5, 6, and 7 compare the results of our method with the ground truth. Figure 5-a plots the average error between the true motion parameters and the estimated ones. One may notice that the average error is kepts within a few degrees, with the exception of frame 98 for which the average error is 15^0. Nevertheless, the tracker was able to recover from this large error. Figure 5-b shows the initial error between the predicted contours and

the image contours (top curve) as well as the error once the motion parameters were fitted using the minimization described in section 5.3. The failure of the tracker at frame 98 corresponds to an error of 4-5 pixels.

In more detail, Figures 6 and 7 compare the estimated trajectories with the ground truth for the left and right elbow and for the left and right shoulder. Both the elbow and shoulder are modeled with two rotational degrees of freedom.

6.2 Tracking real motions

In order to track real human-body motions we used a setup composed of six cameras that were accurately calibrated and whose video outputs are finely synchronized. Fine synchronization (of the order of 10^{-6}s) combined with fast shutter speed (10^{-3}s) allow one to cope with fast motions. The camera setup is shown on Figure 8. It consists in six Firewire cameras that deliver 600×800 uncompressed images at 30 frames per second.

We used two different characters, Ben and Erwan, Figure 9. These characters have the same size and therefore we used the same parameters for the elliptical cones modelling the body parts.

We gathered three sets of data shown on Figure 1 (Erwan-1), Figure 6.2 (Ben) and Figure 6.2 (Erwan-2). For each data set, the figures show the images associated with the first three cameras, the associated silhouettes, and the estimated pose of the model displayed from the viewpoint of the third camera. The extremal contours eventually fitted to the data are shown overlayed onto the raw images. The first example, Erwan-1, contains 250 images per sequence, the second example, Ben, contains 800 images per sequence and the third example, Erwan-2, contains 200 images per sequence. Notice that the Erwan motions involve all the degrees of freedom of the articulated model, as well as as the motion of the root body-part.

In all these examples, the minimization process that stands in the inner loop of the tracker converges in 5 to 7 iterations. The RMS error is generally close to one to two pixels. It is worthwhile to notice that the characters' parameters (dimensions of body parts such as arms, feet, thighs, head, etc.) are roughly estimated and hence the RMS error could be improved with better estimates of the characters' dimensions.

7 Conclusion

In this report we proposed a method for tracking the motion of articulated objects. The main contribution of the report is twofold. First we derive an exact kinematic parameterization of the extremal contours produced by developable surfaces. Second we combine this parameterization with the zero-reference kinematic model that is well suited for representing the space of human motions, i.e., a combination of both the sapce of articulated motions (spanned by rotational joints) and the space of free motions (spanned by three rotations and three translations). We derived an analytical expression for the Jacobian linking joint- and free-motion velocities to extremal-contour velocities and we

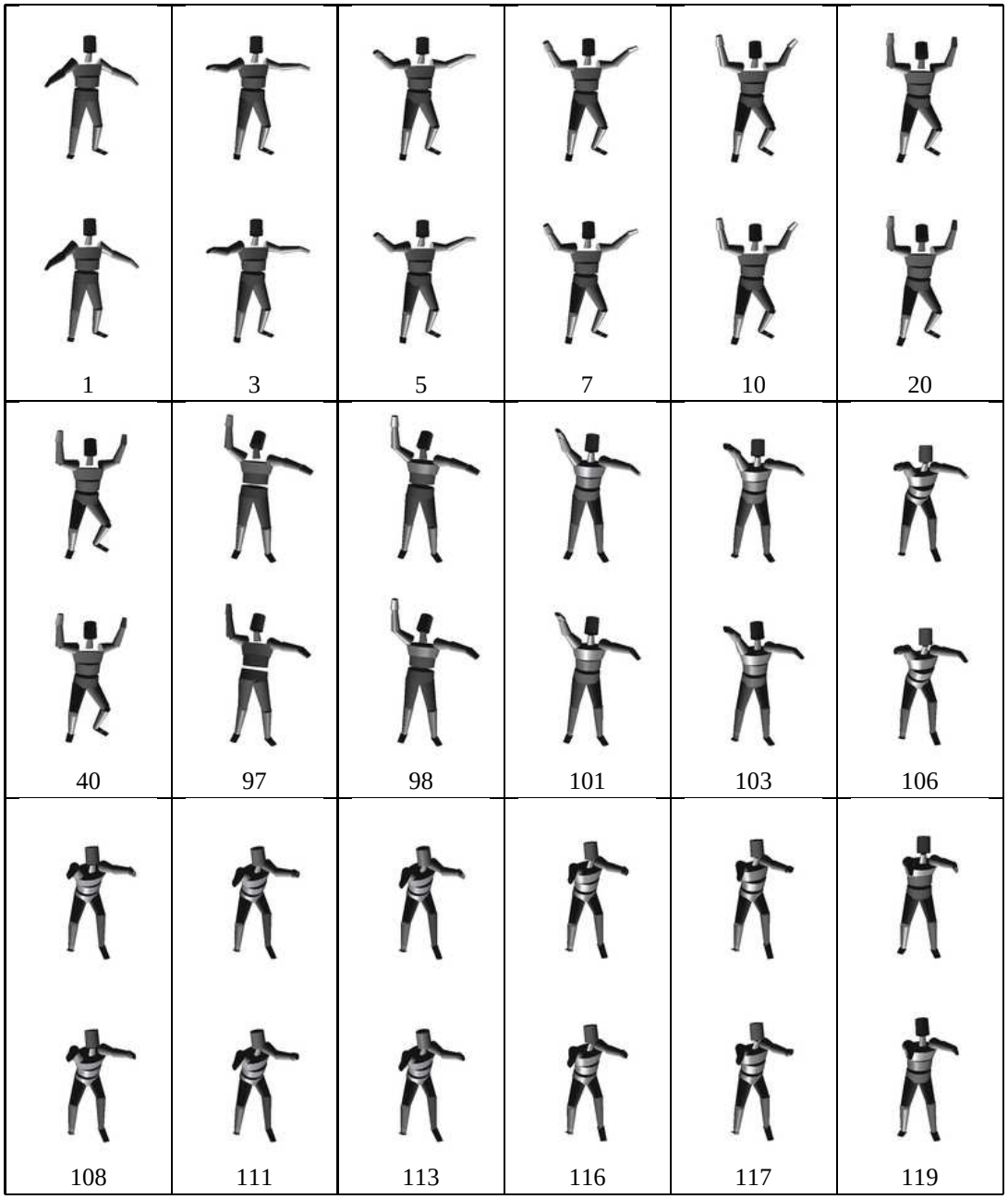
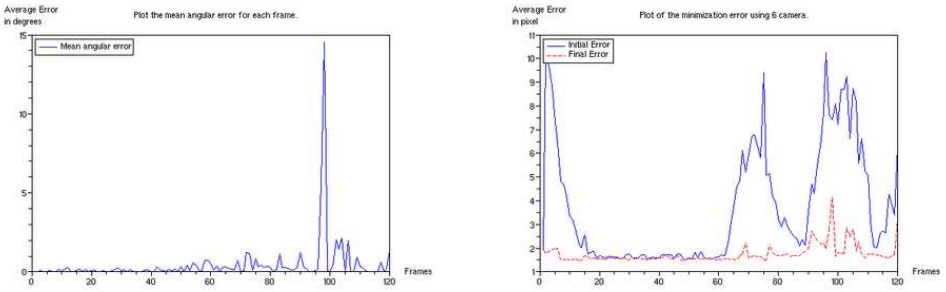


Figure 4: The simulated poses (top) used as the ground truth and from which we generated silhouettes to be used by our tracker. The recovered poses (bottom) using the tracker.



(a) (b)

Figure 5: (a) – The average error between the simulated motion parameters and the estimated ones. (b) – The error between the image contours and the extremal contours before the minimization (top curve) and after the minimization (bottom curve).

showed how this Jacobian matrix can be plugged into a non-linear minimization method that uses the directed chamfer distance between extremal contours predicted by the model and image contours extracted from silhouettes. One major advantage of using the directed chamfer distance is that it does not require one-to-one assignments between image and model contour points.

Moreover, we analysed the conditions under which the minimization process can be carried out effectively. Although, in principle, one camera may suffice, in practice it is desirable to have several images gathered simultaneously with several cameras. We carried out a large number of experiments with both simulated and real data. The tracker performs very well and is able to recover from badly estimated poses.

In the future we plan to investigate ways to combine the directed chamfer distance with a "soft assignment" paradigm that starts with several extremal-contour-to-image-contour hypotheses and iteratively chooses the best assignment.

References

[1] Ankur Agarwal and Bill Triggs. Recovering 3d human pose from monocular images. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 28(1), January 2006.

[2] H. G. Barrow and J. M. Tenenbaum. Interpreting line drawings as three-dimensional surfaces. *Artificial Intelligence*, 17(1-3):75–116, August 1981.

[3] G. Borgefors. Distance transformation in digital images. *Computer Vision, Graphics, and Image Processing*, 34(3):344–371, 1986.

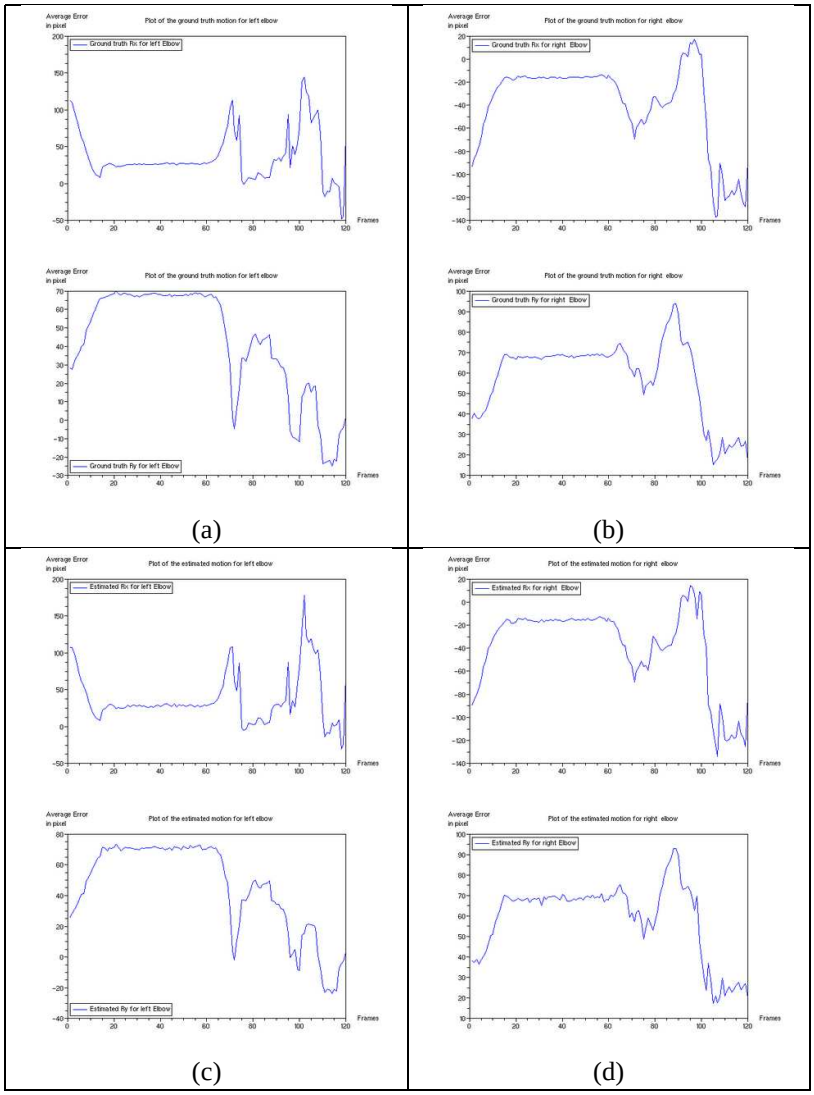


Figure 6: Ground-truth and estimated joint-angle trajectories of the left elbow and of the right elbow.

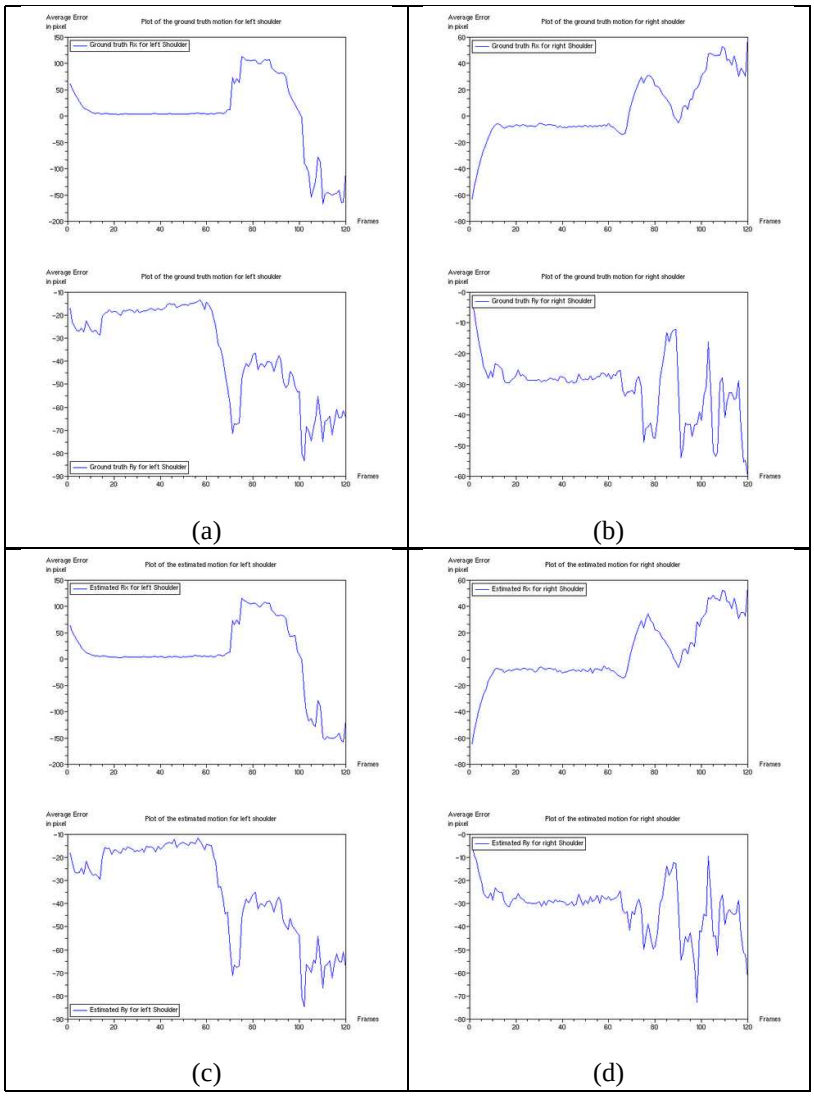


Figure 7: Ground-truth and estimated joint-angle trajectories of the left shoulder and of the right shoulder.

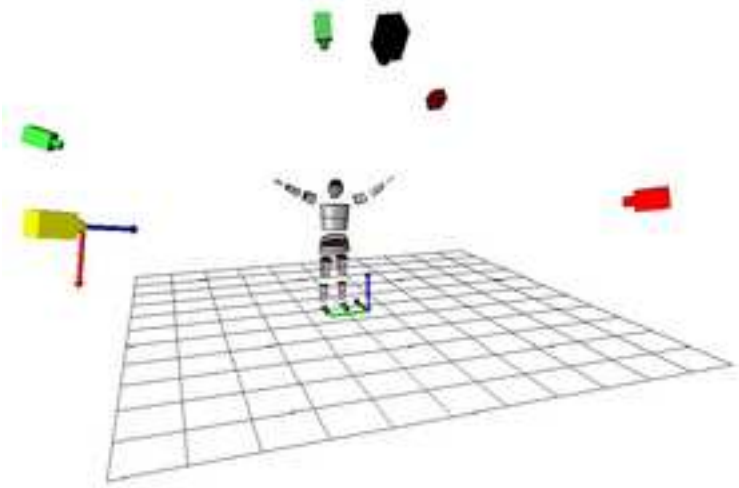


Figure 8: The camera setup used in our experiments. The cameras are calibrated with respect to a global reference frame. The human body model is shown in it's reference position.

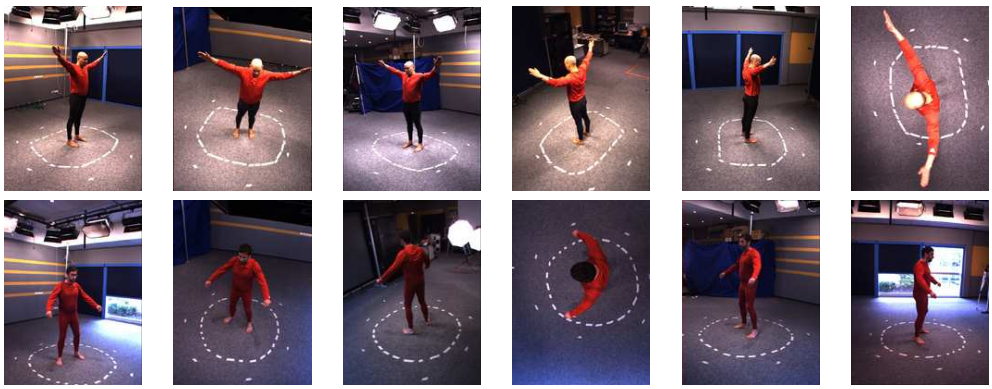
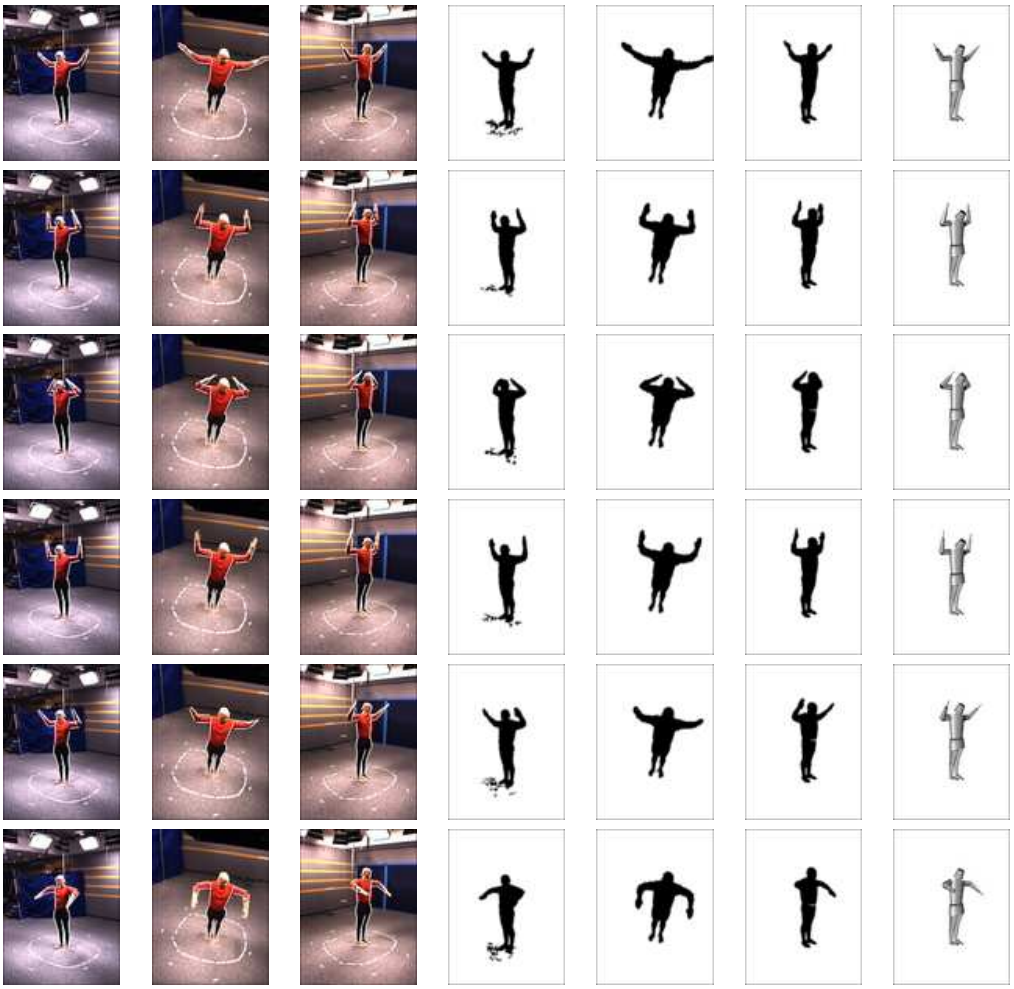


Figure 9: Six view of Ben (top) and of Erwan (bottom).



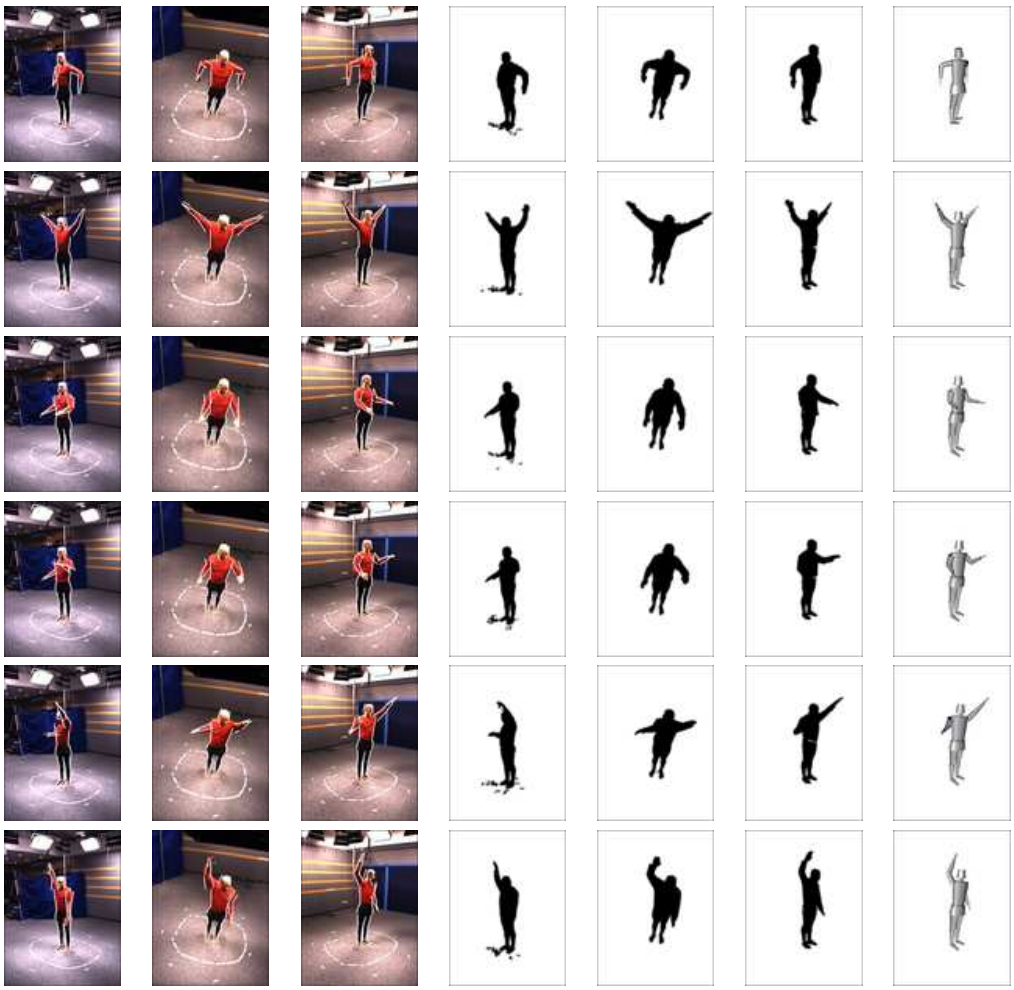


Figure 10: The result of tracking Ben with 6 cameras over 800 images.

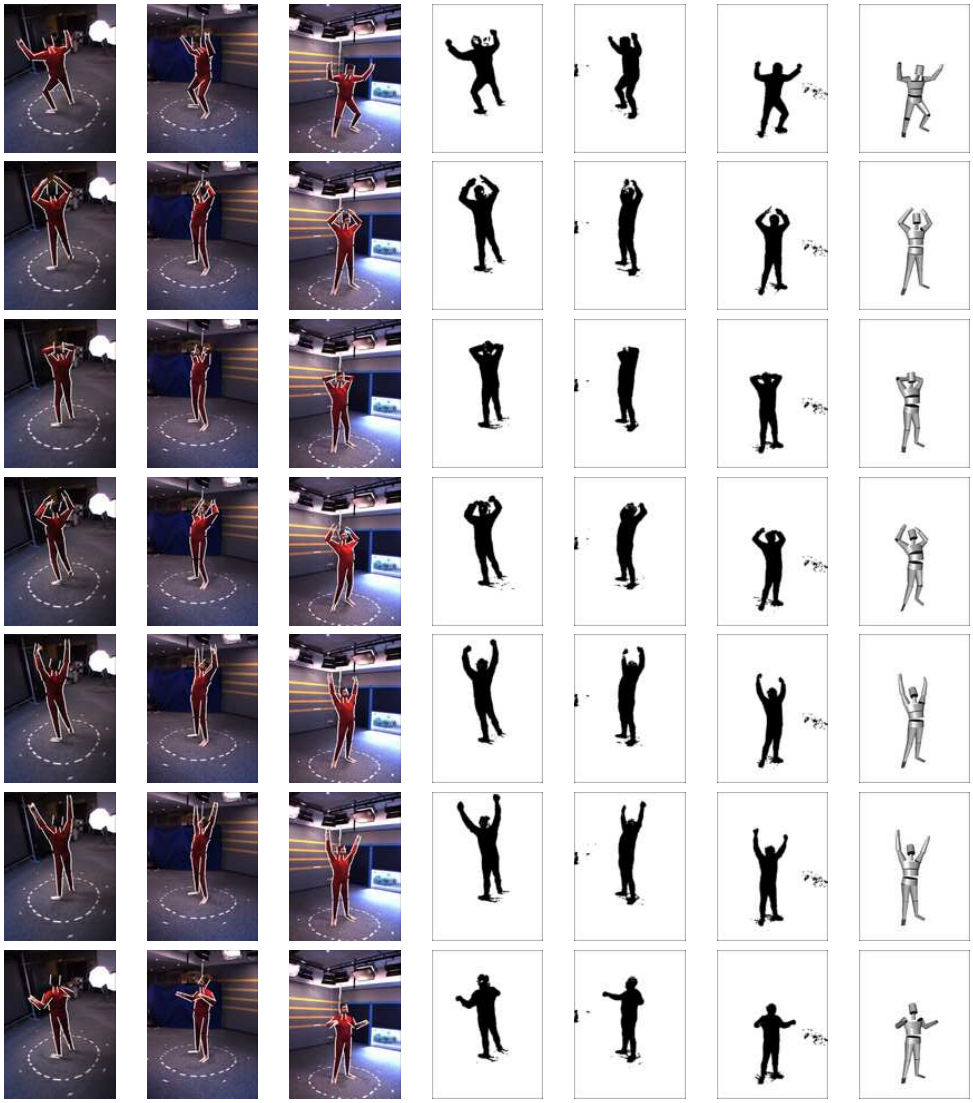


Figure 11: The result of tracking Erwan-2 with 6 cameras over 250 frames.

[4] C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, 56(3):179–194, February - March 2004.

[5] P. David, D.F. DeMenthon, R. Duraiswami, and H. Samet. Softposit: Simultaneous pose and correspondence determination. *International Journal of Computer Vision*, 59(3):259–284, September 2004.

[6] Q. Delamarre and O. Faugeras. 3d articulated models and multi-view tracking with physical forces. *Computer Vision and Image Understanding*, 81(3):328–357, 2001.

[7] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Computer Vision and Pattern Recognition*, pages 2126–2133, 2000.

[8] M. P. Do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

[9] T. Drummond and R. Cipolla. Real-time tracking of highly articulated structures in the presence of noisy measurements. In *ICCV*, pages 315–320, 2001.

[10] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. Pattern Analalysis Machine Intelligence*, 24(7):932–946, 2002.

[11] D.A. Forsyth and J. Ponce. *Computer Vision – A Modern Approach*. Prentice Hall, New Jersey, 2003.

[12] D.M. Gavrila and L.S. Davis. 3d model-based tracking of humans in action: a multi-view approach. In *Conference on Computer Vision and Pattern Recognition*, pages 73–80, San Francisco CA, 1996.

[13] D.M. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *IEEE Proceedings of the Seventh International Conference on Computer Vision*, pages 87–93, Kerkyra, Greece, 1999.

[14] R. Horaud and M. Brady. On the geometric interpretation of image contours. *Artificial Intelligence*, 37(1–3):333–353, December 1988.

[15] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, September 1993.

[16] I. Kakadiaris and D. Metaxas. Model-based estimation of 3d human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1453–1459, 2000.

[17] David Knossow, Remi Ronfard, Radu Horaud, and Frédéric Devernay. Tracking with the kinematics of extremal contours. In *Computer Vision – ACCV 2006*, LNCS, pages 664–673, Hyderabad, India, January 2006. Springer.

[18] J. Koenderinck. *Solid Shape*. The MIT Press, 1990.

[19] F. Martin and R. Horaud. Multiple camera tracking of rigid objects. *International Journal of Robotics Research*, 21(2):97–113, February 2002.

[20] J. M. McCarthy. *Introduction to Theoretical Kinematics*. MIT Press, Cambridge, 1990.

[21] B. W. Mooring, Roth Z. S., and M. R. Driels. *Fundamentals of Manipulator Calibration*. John Wiley & Sons, 1991.

[22] R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Ann Arbor, 1994.

[23] D.G. Sim, O.K. Kwon, and R.H. Park. Object matching algorithms using robust hausdorff distance measures. *IEEE Transactions on Image Processing*, 8(3):425–429, March 1999.

[24] C. Sminchisescu and Bill Triggs. Kinematic jump processes for monocular 3d human tracking. In *International Conference on Computer Vision and Pattern Recognition*, pages I 69–76, June 2003.

[25] C. Sminchisescu and Bill Triggs. Building roadmaps of minima and transitions in visual models. *International Journal of Computer Vision*, 61(1):81–101, 2005.

[26] Y. Song, L. Goncalves, and P. Perona. Unsupervised learning of human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):814–827, July 2003.

[27] K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *International Journal of Computer Vision*, 48(1):9–19, June 2002.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifi que
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399